University of Alberta

Adaptive Process Monitoring and Sensor Validatiqn

by

Abhishek Bhargava

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Master of Science**.

in

Process Control

Department of Department of Chemical and Materials Engineering

Edmonton, Alberta
Spring2001

National Library of Canada
Acquisitions and Bibliographic Services
395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale du Canada
Acquisitions et services bibliographiques
395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-60413-6

Canada

University of Alberta

Library Release Form

Name of Author: Abhishek Bhargava

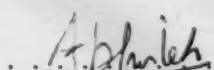
Title of Thesis: Adaptive Process Monitoring and Sensor Validation

Degree: Master of Science

Year this Degree Granted: 2001

Permission is hereby granted to the University of Alberta Library to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as hereinbefore provided, neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's prior written permission.



Abhishek Bhargava
C-19, Lal Bahadur Shastri Marg
Tilak Nagar
Jaipur, Rajasthan
India, 302004

Date: Jan 29, 2001.

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **Adaptive Process Monitoring and Sensor Validation** submitted by Abhishek Bhargava in partial fulfillment of the requirements for the degree of **Master of Science** in *Process Control.*

Dr. S. L. Shah (Supervisor)

Dr. Biao Huang

Dr. Clayton V. Deutsch

Date: Jan 22, 2001.

To my Parents

Abstract

As chemical processes become more and more complex, one of the significant goals in process industries is to increase the safety and reliability of such processes. Significant effort has gone into the development of quality control and process monitoring methodologies. Equally important in the control of chemical processes is a fast and efficient means of detecting and isolating faults.

Multivariable statistical analysis methods, e.g., Principal Component Analysis (PCA), Partial Least Squares (PLS) and Canonical Variate Analysis (CVA) have widely been applied to chemical process monitoring. However, few recursive monitoring techniques have been developed for fully dynamic and time-varying multivariate processes. Recursive PCA has been proposed and successfully applied to the monitoring of static time-varying processes but it is not congenial for fully dynamic processes. Dynamic PCA has been developed, however its recursive variant is not available. Many processes operate in dynamic states and are often time-varying. The time varying property includes not only the variation of parameters but also the process structure.

In this thesis, two main issues are investigated. First, a novel approach to the adaptive monitoring of multivariable dynamic and time-varying processes by the use of recursive multi-channel lattice filter is presented. The lattice filter algorithm is derived from a geometric point of view. The effectiveness of the newly proposed approach is demonstrated on a simulated system and a pilot scale plant, respectively.

Second, detection, identification and reconstruction of faulty sensors in fully dynamic systems is studied. To generate the primary residual vector for fault detection, the error-in-variables (EIV) subspace model is used for the identification of the residual model from the noisy measurement data, without the need of any *priori* knowledge of the system under consideration. To identify the faulty sensor, a dynamic structured residual approach with maximized sensitivity (DSRAMS) is used. In this methodology a set of structured residuals where one residual is most sensitive to one specified subset of faults but immune to others are generated. Fault magnitude is then estimated based on the model and faulty data

and the faulty sensor is reconstructed. DSRAMS is applied on a simulated system for the isolation of single and simultaneous multiple sensor faults. Also DSRAMS has been successfully applied on-line on a pilot scale plant for the isolation of single sensor faults.

Acknowledgements

I would like to express my gratitude to Dr. Sirish L. Shah for his enthusiastic supervision, help and guidance throughout the course of this work. I am grateful to him for his comments and useful insights. I would also like to thank Profs. Shah, Chen and Weins for teaching the fundamentals of control and statistics. I was fortunate to learn the subjects from such fine teachers.

I would like to thank Dr. Weihua Li for his constant help and the useful discussions I had with him. It was my pleasure to work with Dr. Weihua Li for the past one year. A big THANKS to my colleagues in Process Control Group : Ashish, Ramesh, Kamrun, Arun, Bhushan, Lanny, Dongguang, Raghu, Hari and others for helping me in some way or the other. Special thanks to Bhushan for his help in building the Graphical User Interface. I would also like to thank NSERC and the Department of Chemical Engineering for their financial assistance during the course of this MSc research.

I will always remember the good times I had with Sandip, Geetanjali, Raja, Lekhi and Khulbe families. Khulbe and Lekhi families have been very helpful to me over the last two years.

Contents

1 Introduction 1

1.1 Scope of the thesis . . . 2

1.2 Organization of this thesis . . . 3

2 Methods for Process Monitoring and Fault Diagnosis: A survey 4

2.1 Introduction . . . 4

2.2 Principal Component Analysis (PCA) . . . 5

2.2.1 Application . . . 6

2.3 Canonical Correlation Analysis (CCA) . . . 7

2.4 Partial Least Squares (PLS) . . . 9

2.5 Kalman Filter based Residual Generation . . . 10

2.6 The Local Approach . . . 10

2.7 Summary . . . 12

3 Adaptive process monitoring via multi channel Lattice Filter 13

3.1 Introduction . . . 13

3.2 Lattice Filter . . . 14

3.2.1 Mathematical Preliminaries . . . 15

3.2.2 Derivation of Lattice filter . . . 17

3.3 Problem Formulation . . . 20

3.4 Residual generation using multi-channel Lattice filter . . . 24

3.4.1 Notations and definitions . . . 24

3.4.2 Associating the Lattice Filter with Parameters $\theta_n^i(k)$ . . . 27

3.5 Lattice filter Algorithm . . . 28

3.5.1 Order Recursion Equations . . . 28

3.5.2 Time Recursion Equations . . . 31

3.5.3 Parameter Recursion Equations . . . 34

3.6 Lattice filter based Process Monitoring . . . 35

3.6.1 Initialization . . . 35

3.6.2 Monitoring Index . . . 35

3.6.3 Treatment of Some Practical Issues . . . 37

3.6.4 Adaptive Process Monitoring 40
3.7 Case Studies 40
3.7.1 Case Study One- Simulation Example 41
3.7.2 Case Study Two- Pilot scale plant 44
3.8 Conclusions 47

4 Sensor validation **51**
4.1 Introduction 51
4.2 Residual Generation and Fault Detection Index 53
4.2.1 Fault Representation and Residual Generation 53
4.2.2 Fault Detection Index 56
4.3 Fault Identification with DSRAMS 57
4.3.1 Concept of Structured Residuals 57
4.3.2 Detectability, Attainability and Isolability 59
4.3.3 Dynamic Structured Residual Approach with Maximized Sensitivity (DSRAMS) 60
4.4 Fault Identification Indices 61
4.4.1 Filtered Structured Residuals (FSR) 62
4.4.2 Cumulative Sum (Q_{sum}) 62
4.4.3 Cumulative Variance (V_{sum}) 63
4.5 Reconstruction of Faulty Sensors 63
4.6 Case Studies 64
4.6.1 Case Study One - Simulation Example 64
4.6.2 Case Study Two - Pilot Scale Plant 75
4.7 Summary 76

5 Summary and Conclusions **85**
5.1 Directions for Future Work 86

Bibliography **87**

A **92**
A.1 Derivation of Equation 3.37 92
A.2 Derivation of Equation 3.39 94
A.3 Derivation of Equations 3.43 and 3.44 94
A.3.1 Derivation of Equation 3.43 95
A.3.2 Derivation of Equation 3.44 96
A.4 Proof of Equation 3.46 96
A.5 The Complete Algorithm for the Recursive Identification of $\Theta_n^i(k)$ 97

List of Figures

2.1 Hotelling T^2 plot for process monitoring 8
2.2 PC 10 Vs PC 1 Scores plot 8

3.1 Illustration of projection theorem 15
3.2 Realization of m-th order predictors 18
3.2.1 Forward predictor 18
3.2.2 Backward predictor 18
3.3 Lattice filter based n-th order predictor 20
3.3.1 Overall structure 20
3.3.2 Single stage 20
3.4 T^2 statistic for the Constant Model-based Process Monitoring on the Simulated System 43
3.5 T^2 statistic for the Adaptive Lattice Filter-based Process Monitoring on the Simulated System 44
3.6 Tracking the Parameters of the AR Part of the Simulated System 45
3.7 Schematic diagram of the CSTHS 46
3.8 Flow Rates of Steam and Cold Water; Level and Temperature of Cold Water in CSTHS 48
3.9 T^2 statistic for the Constant Model-based Process Monitoring on CSTHS (using every $4th$ sample period) 49
3.10 T^2 statistic for the Adaptive Lattice Filter-based Process Monitoring on CSTHS (using every $4th$ sample period) 49
3.11 Order Determination of the CSTHS Model 50

4.1 Precision degradation fault in sensor 1 67
4.2 Bias fault in sensor 2 68
4.3 Drift fault in sensor 6 69
4.4 Complete failure fault in sensor 7 70
4.5 Simultaneous sensor fault detection: Drift in sensor 1 and bias in sensor 7 . 71
4.6 Simultaneous sensor fault detection: Complete failure in sensor 3 and precision degradation in sensor 6 72

4.7 Simultaneous sensor fault detection: Bias in sensor 4 and precision degradation in sensor 5 73
4.8 Simultaneous sensor fault detection: Drift in sensor 2 and complete failure in sensor 7 74
4.9 Bias in sensor 1 (level) 77
4.10 Precision degradation in sensor 2 (temperature) 78
4.11 Complete failure in sensor 3 (Steam valve) 79
4.12 Drift in sensor 4 (Cold water valve) 80
4.13 Bias in sensor 1 (level) 81
4.14 Complete failure in sensor 2 (temperature) 82
4.15 Bias in sensor 3 (Steam valve) 83
4.16 Precision degradation in sensor 4 (Cold water valve) 84

Chapter 1

Introduction

On-line process monitoring, fault detection and diagnosis (FDD) of industrial processes is crucial to operational safety and monitoring product quality. The early indication of incipient failures can help avoid major plant breakdowns and catastrophies, ones that could otherwise result in substantial material damage and even human fatalities. Similarly, fault detection and isolation has become a critical issue not only in chemical process industries but also in the operation of high-performance systems such as ships, submarines, air planes and space vehicles, where safety and significant material value are at stake. The faults are basically classified into ***additive faults*** and ***multiplicative faults***. Sensor and actuator faults are the typical additive faults as they do not affect the model of the process under consideration. Process faults are typical multiplicative faults as they affect the process model parameters. A typical procedure for process monitoring, sensor fault detection and diagnosis consist of three steps: (1) residual generation, (2) analysis of residuals for fault isolation and (3) reconstruction of faulty sensors or actuators. The first two steps are also referred to as fault detection and isolation (FDI), in which the residual generation is the most important step. The residuals of a process are the linear combination of outputs and inputs. To generate residuals for process monitoring and diagnosis, an adequate knowledge either of the process or system identification techniques is essential.

A process model can be obtained from first principles if the mechanism is simple and fully understood. However, as most industrial processes are quite complicated, the process model is often obtained by using system identification techniques from the sampled input and output data. System identification, thus, plays an important role in process monitoring and FDD. A number of models structures such as transfer function model with: AR (auto-regressive), ARMA (auto-regressive moving average), ARMAX (auto-regressive moving average exogenous) and FIR (finite impulse response) structures or state-space models may be identified.

Most industrial processes, e.g., chemical processes, are multivariate in nature. When using classical system identification techniques to build the above mentioned models, the process variables are divided into two groups: the inputs and the outputs. Also it is

implicitly assumed that there is some cross correlation between the two groups and that the variables are independent within each group. Since processes are heavily instrumented, the sampled variables within a group can be dependent and such a dependency can cause severe numerical problem for the model estimation. In fact only few variables are independent, called latent variables, which can basically characterize a process. To overcome the problems of linear dependency or redundancy among the entire sampled variables, latent variables based system identification techniques, e.g., Principal component analysis (PCA), Canonical variate analysis (CVA) and Partial least square (PLS), have been widely applied to chemical processes for monitoring and diagnosis (Larimore 1983, MacGregor 1989, Kresta *et al.* 1991, Piovoso *et al.* 1992, Schaper *et al.* 1994, Kourti and MacGregor 1995, Lakshminarayanan *et al.* 1995, Wise *et al.* 1996, Dunia and Qin 1998). Recently, the subspace methods of identification (SMI) have become very popular among the system identification community and have been widely applied towards modeling and monitoring of industrial processes (Larimore June 1997, Basseville *et al.* 2000).

1.1 Scope of the thesis

Many industrial processes operate in dynamic state and are often time-varying. The time-varying property includes not only the variation of parameters but also the variation of process structure, e.g., the change of the process order. However, for such dynamic and time-varying processes, few recursive techniques are available. Recursive PCA (W. Li and Qin 2000) has been proposed and successfully applied to quasi-steady state process monitoring but it does not work for fully dynamic processes. Dynamic PCA has also been developed for process diagnosis (Ku *et al.* 1995), however its recursive variant is not yet available.

In this thesis a novel approach to the adaptive monitoring of multivariate dynamic time-varying processes via the use of multi-channel lattice filters is presented. Lattice filters have been extensively applied to signal processing and system identification (Lee *et al.* 1981, Friedlander 1982, Kummert *et al.* 1992). Their main advantage is that they are recursive both in time and in order. They also have the attribute of allowing fast convergence and good tracking of time-varying process variables (Honig and Messerschmitt 1981, Lev-Ari *et al.* 1984).

It is believed that errors-in-variables (EIV) state space model can represent a multivariate dynamic process in the presence of process noise and measurement noise in the inputs and the outputs. The identification algorithm for the EIV state space model has been proposed by Chou and Verhaegen (1997), and can be directly applied to residual generation for process monitoring and fault detection (Qin and Li 1999). However, this algorithm is based on the assumption that the process is time-invariant. More recently, Gustafsson proposed a recursive version of the algorithm (Gustafsson *et al.* 1998). However this algorithm is only recursive in time. By showing the relationship between the process EIV state space representation and the multi channel lattice filter, the lattice filter is used to generate a

residual vector for process monitoring. By making use of the lattice filter's ability of recursively updating the process model both in time and order, an on-line algorithm to update the residual vector with the newly sampled process data is proposed. A practical approach to determine the process model order is also included. Hotelling T^2 statistic, constructed from the sequence of residual vectors, is used as the monitoring index.

The second part of the thesis is concerned with the sensor or actuator fault detection, isolation and reconstruction in ***fully dynamic processes***. A subspace model is obtained by the use of the EIV system identification algorithm to generate residuals for fault detection. Further, for the isolation of faulty sensors, a ***dynamic structured residual approach with maximized sensitivity*** (DSRAMS) is used to generate a set of structured residuals where one residual is designed to be most sensitive to a given subset of faults but immune to others (Qin and Li 2000). After the isolation, the fault magnitude is estimated based on the model and the faulty data, and the faulty sensors are reconstructed.

1.2 Organization of this thesis

As outlined in the earlier section, Chapter 2 starts with a brief overview of various existing techniques in the area of multivariate statistical analysis based process monitoring and fault detection and diagnosis. Chapter 3 is associated with the lattice filter based multivariate time-varying dynamic process monitoring, starting with a brief introduction of the lattice filter. After showing the equivalence between the lattice filter and a multivariate dynamic process, a detailed derivation of the lattice filter algorithm, including the time and order update equations, is presented from a geometric point of view. Further, a lattice filter based process monitoring scheme is proposed and two case studies are presented to show the effectiveness of the proposed scheme for adaptive process monitoring.

Chapter 4 deals with the detection and isolation of faulty sensors in fully dynamic systems. Applications of DSRAMS are illustrated for single and simultaneous multiple sensor faults. Finally the thesis ends with conclusions and directions for future work in Chapter 5.

Chapter 2

Methods for Process Monitoring and Fault Diagnosis: A survey

2.1 Introduction

Advances in the areas of instrumentation and data acquisition have made it possible to collect large amount of data in process industry. Previously, univariate statistical process control (SPC) charts such as Shewhart charts, Exponential weighted moving average (EWMA) trends, Cumulative sum (CUSUM) etc., were applied to monitoring key process variables in multivariate processes in order to detect the occurrence of abnormal conditions, where interaction among different variables was not taken into account. As most chemical processes are multivariate by nature, the use of univariate SPC charts not only result in misleading process information but also makes the interpretation and diagnosis difficult.

Recently, multivariate statistical methods such as PCA, PLS and CCA are finding increased use in the analysis of multivariate processes. MacGregor and his coworkers (Kresta 1992, Nomikos and MacGregor 1995) and Wise (Wise 1991) have done pioneering work in applying PCA and PLS to monitoring chemical processes. The usefulness of PCA and PLS for preliminary data analysis, process monitoring and control system designing has also been demonstrated on the Shell Standard Control Problem (Kasper and Ray 1992). The use of EWMA filters in conjunction with PCA and PLS has been discussed in Wold (1994). A multi-way PCA approach to monitoring batch processes has been reported by Nomikos and MacGregor (1995) and an extension to continuous processes has been made by Chen and McAvoy (1998). Non-linear PCA (Shao *et al.* 1999) has been proposed for process monitoring, where the data is first pre-processed to remove noise and spikes through wavelet de-noising. Process monitoring by Multiscale PCA, where wavelet analysis is used to extract deterministic features, is reported in Bakshi (1998).

Studies bases on PCA and PLS based process monitoring and diagnosis are also presented in Wold et al. (1987), Kourti et al. (1995) and Wise et al. (1996). The use of PCA scores, that are recursively summed up, for process monitoring and enhanced disturbance resolution is presented in Wachs et al. (1999). Application of PCA to high frequency

information, obtained from wavelet decomposition of a signal, to detect a faulty sensor is reported in Luo et al. (1999). Also, the feasibility of sensor fault detection using noise analysis, with the help of power spectrum density estimation and PCA, is carried out in Ying et al. (2000). The use of CCA for chemical process modeling and fault detection applications is reported in Schaper et al. (1994), Lakshminarayanan et al. (1995) and Wang et al. (1997).

While the PCA, PLS and CCA are mainly used for process monitoring, significant work regarding the detection and isolation of faults has been done in terms of the analytical models of the system. Beard (1971) and Jones (1973) have proposed the use of fault detection filter where a fault is detected when one or more of the residual projections along the known fault direction are sufficiently large. A Dedicated observer approach has been used where the outputs of the system are reconstructed from the measurements with the aid of Kalman filters. The estimation error or innovation are used as a residual for the detection and isolation of faults (Clark *et al.* 1975, Mehra and Peshon 1971). Isermann (1984) has proposed the use of parameter identification approach where the faults are detected via the estimation of parameters of the mathematical model. The Parity state space approach has been used where the consistency of mathematical equations of the system is checked by using the actual measurements (Frank 1990, Gertler and Singer 1990). The statistical local approach, which is more suitable for process faults, has also been proposed for detecting parametric changes (Benveniste *et al.* 1987, Basseville and Nikiforov 1993). Input-Training Neural Networks have also been used for dimensionality reduction allowing process monitoring tasks such as missing sensor replacement, sensor error detection and rectification (Reddy and Mavrovouniotis 1998).

Multivariate statistical analysis approaches, e.g., PCA, PLS and CCA and the analytical redundancy approach seem to belong to two different categories. Recently, after investigating the relationship among the two frameworks, Gertler et al. (1999) proposed the isolation enhanced PCA, combining the advantages of PCA for multivariate process monitoring and the analytical redundancy for fault isolation together.

In the following sections, the basic concepts of PCA, PLS and CCA are briefly outlined which is followed by a brief review of some analytical redundancy based approaches for FDI.

2.2 Principal Component Analysis (PCA)

The goal of PCA is to model a single block of data, $\mathbf{X} \in \Re^{N \times p}$ with N observations and p variables, using orthogonal components. It is concerned with explaining the variance-covariance structure through a few *linear* combinations of the original variables. The total variability of the data set is explained by the $p - variable$ space. However if the original variables are correlated, then most of the variability in the data set can be summarized by a lower $n - dimensional$ subspace ($n \ll p$). The n principal components (also called latent variables) can then replace the initial p variables and be used for further analysis purposes.

Consider the following p linear combinations (Johnson and Wichern 1998)

$$\begin{aligned} \mathbf{t}_1 &= l_{11}\mathbf{x}_1 + l_{12}\mathbf{x}_2 + \cdots + l_{1p}\mathbf{x}_p \\ \mathbf{t}_2 &= l_{21}\mathbf{x}_1 + l_{22}\mathbf{x}_2 + \cdots + l_{2p}\mathbf{x}_p \\ &\vdots \\ \mathbf{t}_p &= l_{p1}\mathbf{x}_1 + l_{p2}\mathbf{x}_2 + \cdots + l_{pp}\mathbf{x}_p \end{aligned} \tag{2.1}$$

where $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p$ are the p variables of the data matrix $\mathbf{X}$. Also

$$\begin{aligned} Var(\mathbf{t}_i) &= \mathbf{l}_i^T \mathbf{\Sigma}\, \mathbf{l}_i \qquad i = 1, 2, \ldots, p \\ Cov(\mathbf{t}_i, \mathbf{t}_k) &= \mathbf{l}_i^T \mathbf{\Sigma}\, \mathbf{l}_i \qquad i, k = 1, 2, \ldots, p \end{aligned} \tag{2.2}$$

where $\mathbf{\Sigma}$ is the covariance matrix of $\mathbf{X}$. The principal components are those *uncorrelated* linear combinations $\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_p$ whose variances are as large as possible. The first principal component, $\mathbf{t}_1$, has the maximum variance, the second principal component, $\mathbf{t}_2$, accounts for the maximum variance that has not been accounted for by $\mathbf{t}_1$, and so on. The following constraints

$$\mathbf{l}_i^T \mathbf{l}_i = 1 \quad (i = 1, \cdots, p)$$

and

$$\mathbf{l}_i^T \mathbf{l}_k = 0 \quad (i \neq k)$$

are placed to achieve the above objective.

The principal component weights matrix, $\mathbf{P}$ which consists of weight vectors $\mathbf{l}_1$ to $\mathbf{l}_n$, is obtained by performing singular value decomposition (SVD) on the data matrix $\mathbf{X}$ (Lakshminarayanan 1997). In the presence of correlation among the variables, some of the singular values and the corresponding principal components will be insignificant. Noise and redundancies present in the data are contained to these insignificant PCA dimensions (from $n+1$ to p). PCA can also be considered as a technique which decomposes the data matrix $\mathbf{X}$ as follows:

$$\mathbf{X} = \mathbf{t}_1\mathbf{l}_1^T + \mathbf{t}_2\mathbf{l}_2^T + \cdots + \mathbf{t}_n\mathbf{l}_n^T + \mathbf{E}_{n+1} = \mathbf{T}\mathbf{P}^T + \mathbf{E}_{n+1} \tag{2.3}$$

In the above equation, the matrix $\mathbf{P}$ is the *loading matrix* which consists of weights associated to the original variables in the construction of the principal components. Also, matrix $\mathbf{T}$ is the *scores matrix* which consists of sample projections on to the lower n-dimensional subspace. Matrix $\mathbf{E}$ is the error matrix which consist of the insignificant information in the data set.

2.2.1 Application

The PCA model, built from the data collected during normal plant operation, can be used for online process monitoring, fault detection and diagnosis. The tools to achieve the objective are the score plots and loading plots.

Score Plots

A score plot is obtained by plotting any two principal component scores for the corresponding PCA dimension, e.g., $\mathbf{t}_1$ vs $\mathbf{t}_2$. It is used to find relationships between the data points. Points that fall close to each other represent similar data points. Under normal operating conditions, the points in the score plots lie within a control limit contour which is an ellipse. When the points begin to fall outside the normal operating region, it is an indication of abnormal shift in the process variables. The score plots can thus be used to detect abnormal process behaviour. However, the score plots are unable to detect the faults if there is no abnormal shift in the process variables and yet the correlation structure break downs. To overcome this, the squared prediction error (SPE) statistic can be used for process monitoring. The SPE can be considered as a measure of the plant-model mismatch. A large value of SPE indicates that the PCA model no longer represents the current plant correlation structure.

Loading Plots

A loading plot is obtained, in a similar manner as the score plot, by plotting any two loading vectors. The loading plot is used to find relationships between the original variables. All variables sharing the same information are clustered together in a loading plot. Loading plots help in visualizing the cluster of variables responsible for any unusual event.

Figures 2.1 and 2.2 show the results of application of Dynamic PCA for process monitoring. Dynamic PCA model is used to monitor a slowly time varying 4 input - 4 output simulated second order dynamic system (refer Section 3.7.1 for the details of the simulated system). Figure 2.1 is the T^2 plot which shows a number of false alarms as the Dynamic PCA model, which consist of 10 principal components, is unable to track the slowly time varying system. Figure 2.2 is the scores plot obtained by plotting principal component PC 10 against PC 1. Even though the process is operating normally, a number of points lie outside the normal operating region triggering false alarms. The same slowly time varying system is monitored via the use of multi channel lattice filter and comparative results are presented in Chapter 3.

2.3 Canonical Correlation Analysis (CCA)

Canonical correlation analysis is a powerful technique to identify and quantify the associations between two set of variables, say $\mathbf{X}$ and $\mathbf{Y}$. Here $\mathbf{X}$ can be considered as an independent block and $\mathbf{Y}$ as the dependent block of data. In canonical correlation analysis, the goal is to relate the *linear combination* of the variables in set $\mathbf{X}$ and $\mathbf{Y}$. The idea is first to determine the pair of linear combinations having the largest correlation. Next, the pair of linear combinations having the largest correlation among all the pairs uncorrelated with the initially selected pair is determined. The pairs of linear combinations are called the

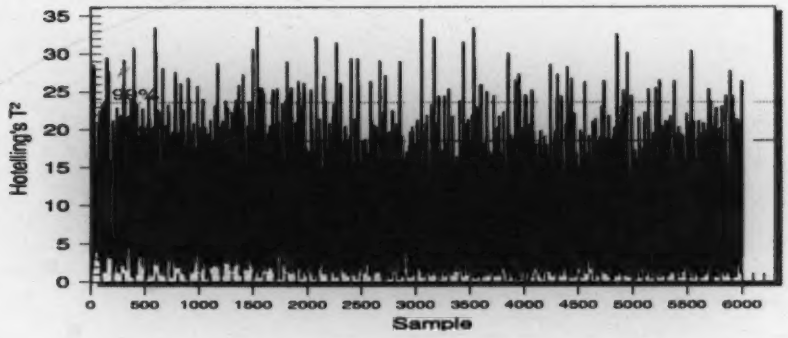


Figure 2.1: Hotelling T^2 plot for process monitoring

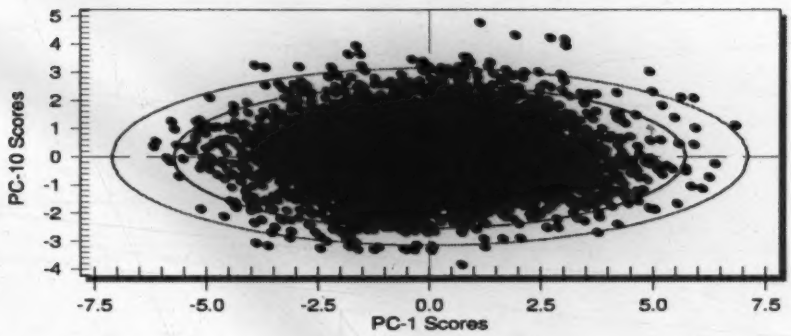


Figure 2.2: PC 10 Vs PC 1 Scores plot

canonical variables and the correlations are called *canonical correlations*. In order to obtain the solution for the required linear combinations, the variance of the canonical variates is constrained to unity. The mathematical details of CCA algorithm are presented in Johnson and Wichern (1998).

2.4 Partial Least Squares (PLS)

Partial least squares is used as an algorithm for computing a solution $\mathbf{B}$ for the regression coefficients in a linear model

$$\mathbf{Y} = \mathbf{XB} + \mathbf{E}$$

where $\mathbf{X}$ and $\mathbf{Y}$ are the known data matrices and $\mathbf{E}$ is the noise. The purpose of using the PLS algorithm for computing $\mathbf{B}$ is to handle the multicollinearity problems where there are linear dependencies among the columns of $\mathbf{X}$ matrix.

Partial least squares has a conceptual similarity to canonical correlation analysis in that the relationship between two blocks of data are modeled to find highly correlated combination of variables. The principal components or the latent variables for the $\mathbf{X}$ block are constructed with reference to the $\mathbf{Y}$ block. The goal of PLS is to form components that capture most of the information in the variables of $\mathbf{X}$ block that is useful for predicting the $\mathbf{Y}$ block variables while reducing the dimensionality of the regression problem by using fewer components than the number of variables in $\mathbf{X}$ block.

In PLS, the input ($\mathbf{X}$) and output ($\mathbf{Y}$) data is expressed as a sum of series of rank 1 matrices as follows:

$$\begin{aligned} \mathbf{X} &= \mathbf{t}_1\mathbf{p}_1^T + \mathbf{t}_2\mathbf{p}_2^T + \cdots + \mathbf{t}_n\mathbf{p}_n^T + \mathbf{E}_{n+1} = \mathbf{TP}^T + \mathbf{E}_{n+1} \\ \mathbf{Y} &= \mathbf{u}_1\mathbf{q}_1^T + \mathbf{u}_2\mathbf{q}_2^T + \cdots + \mathbf{u}_n\mathbf{q}_n^T + \mathbf{F}_{n+1} = \mathbf{UQ}^T + \mathbf{F}_{n+1} \end{aligned} \tag{2.4}$$

where n is the number of PLS dimensions, $\mathbf{E}_{n+1}$ and $\mathbf{F}_{n+1}$ are the residual portion of the input and output data respectively. In the above equations, $\mathbf{T}$ and $\mathbf{U}$ represent the score matrices or the latent variables while $\mathbf{P}$ and $\mathbf{Q}$ represent the loading matrices. The first set of loading vectors is obtained by maximizing the covariance between $\mathbf{X}$ and $\mathbf{Y}$. Projection of the data onto these vectors gives the first score vectors, $\mathbf{t}_1$ and $\mathbf{u}_1$. Matrices $\mathbf{X}$ and $\mathbf{Y}$ are related through their scores by the inner relation which is just a linear regression of $\mathbf{t}_1$ on $\mathbf{u}_1$ giving

$$\hat{\mathbf{u}}_1 = \mathbf{t}_1 b_1$$

In the above equation $\hat{\mathbf{u}}_1\mathbf{q}_1^T$ can be interpreted as the part of $\mathbf{Y}$ data that has been predicted by the first PLS dimension and in doing so the $\mathbf{t}_1\mathbf{p}_1^T$ of $\mathbf{U}$ has been utilized. This process is repeated until all necessary components have been calculated based on the portion of the covariance between $\mathbf{X}$ and $\mathbf{Y}$ that is explained. The PLS based regression model can be

written as

$$\hat{\mathbf{Y}} = \mathbf{TBQ}^T + \mathbf{F}_{n+1}$$

$$\text{where } \mathbf{B} = diag(b_1, b_2, \cdots, b_n).$$

The PLS model obtained from the data collected during normal plant operation, can be used for online process monitoring, fault detection and diagnosis in the same way as the PCA model is used.

2.5 Kalman Filter based Residual Generation

The Kalman filter was originally proposed for the estimation of the state of the system having known system matrices $\{\mathbf{A}(k), \mathbf{B}(k), \mathbf{C}(k), \mathbf{D}(k)\}$ and covariance matrices $\mathbf{R}_o(k)$ and $\mathbf{R}_p(k)$ of the measurement and process noise. Consider the following state space model:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{u}(k) + \mathbf{p}(k) \\ \mathbf{y}(k) &= \mathbf{C}(k)\mathbf{x}(k) + \mathbf{D}(k)\mathbf{u}(k) + \mathbf{o}(k) \end{aligned}$$

where $\mathbf{p}(k)$ and $\mathbf{p}(k)$ are assumed to be Gaussian distributed white noise. The Kalman filter algorithm is as follows

$$\begin{aligned} \hat{\mathbf{x}}(k+1|k) &= \mathbf{A}(k)\hat{\mathbf{x}}(k|k) + \mathbf{B}(k)\mathbf{u}(k) \\ \hat{\mathbf{x}}(k|k) &= \hat{\mathbf{x}}(k|k-1) + \mathbf{K}(k)\boldsymbol{\varepsilon}(k) \end{aligned}$$

where $\mathbf{K}(k)$ is the Kalman filter gain matrix which can be recursively computed and $\boldsymbol{\varepsilon}(k)$ is the *innovation sequence* which is computed as

$$\boldsymbol{\varepsilon}(k) = \mathbf{y}(k) - \mathbf{D}(k)\mathbf{u}(k) - \mathbf{C}(k)\hat{\mathbf{x}}(k|k-1)$$

It can be shown that the innovation $\boldsymbol{\varepsilon}(k)$, given by above equation, is a Gaussian distributed zero mean random vector under normal operating situation. However, in the presence of sensor or actuator faults the mean of the innovation deviates from zero. Hence the occurrence of faults can be detected by monitoring the mean of the innovation sequence.

2.6 The Local Approach

The local approach was first proposed to detect parametric changes in single-input single-output (SISO) AR processes (Benveniste *et al.* 1987). It has been later extended for detecting additive and multiplicative faults (Basseville and Nikiforov 1993) in multivariate processes. The core of this approach is the reduction of fault detection and isolation problem to that of monitoring the mean of a sufficient statistic vector. The basic idea of the local approach is presented below for an AR process.

Consider the following scalar AR process

$$y(k) - a_1^* y(k-1) - \cdots - a_n^* y(k-n) = \epsilon(k)$$

where $\epsilon(k)$ is Gaussian distributed white noise with variance σ^2. Denoting

$$\boldsymbol{\theta}^* \equiv [a_1^* \; a_2^* \; \cdots \; a_n^*]^T \in \Re^n$$

as a vector consisting of true parameters of the process and

$$\boldsymbol{\phi}(k) = [y(k-1) \; y(k-2) \; \cdots \; y(k-n)]^T \in \Re^n$$

as the regressor, the AR process equation is rewritten as

$$y(k) = \boldsymbol{\phi}^T(k)\boldsymbol{\theta}^* + \epsilon(k)$$

As $\boldsymbol{\theta}^*$ is usually unknown, it is replaced by the estimate $\boldsymbol{\theta}_0$ and the prediction error then is

$$\begin{aligned} \hat{\epsilon}(k) &= y(k) - \boldsymbol{\phi}^T(k)\boldsymbol{\theta}_0 \\ &= \boldsymbol{\phi}^T(k)\left(\boldsymbol{\theta}^* - \boldsymbol{\theta}_0\right) + \epsilon(k) \end{aligned}$$

If there is a sudden change in some elements of $\boldsymbol{\theta}_0$, then

$$\Delta\boldsymbol{\theta} = (\boldsymbol{\theta} - \boldsymbol{\theta}^*)\sqrt{N}$$

where N is the length of training data used for identifying $\boldsymbol{\theta}_0$, and the prediction error becomes

$$\hat{\epsilon}(k) = \boldsymbol{\phi}^T(k)\frac{\Delta\boldsymbol{\theta}}{\sqrt{N}} + \boldsymbol{\phi}^T(k)\left(\boldsymbol{\theta}^* - \boldsymbol{\theta}_0\right) + \epsilon(k)$$

It should be noted that $\Delta\boldsymbol{\theta}$ can contain one (single fault) or more than one (multiple faults) non-zero element. In order to detect such a parametric change, the following terms are defined:

$$\begin{aligned} \mathrm{K}\left(\boldsymbol{\theta}, \boldsymbol{\theta}_0, \boldsymbol{\phi}(k)\right) &= \boldsymbol{\phi}(k)\hat{\epsilon}(k) \\ &= \boldsymbol{\phi}(k)\boldsymbol{\phi}^T(k)\frac{\Delta\boldsymbol{\theta}}{\sqrt{N}} + \boldsymbol{\phi}(k)\boldsymbol{\phi}^T(k)\left(\boldsymbol{\theta}^* - \boldsymbol{\theta}_0\right) + \boldsymbol{\phi}(k)\epsilon(k) \in \Re^n \end{aligned} \quad (2.5)$$

and

$$\begin{aligned} \boldsymbol{\xi}_N(\boldsymbol{\theta}) &= \frac{1}{\sqrt{N}}\sum_{k=1}^{N}\mathrm{K}\left(\boldsymbol{\theta}, \boldsymbol{\theta}_0, \boldsymbol{\phi}(k)\right) \\ &= \frac{1}{N}\sum_{k=1}^{N}\left(\boldsymbol{\phi}(k)\boldsymbol{\phi}^T(k)\right)\Delta\boldsymbol{\theta} + \frac{1}{\sqrt{N}}\sum_{k=1}^{N}\left(\boldsymbol{\phi}(k)\boldsymbol{\phi}^T(k)\right)\left(\boldsymbol{\theta}^* - \boldsymbol{\theta}_0\right) + \frac{1}{\sqrt{N}}\sum_{k=1}^{N}\boldsymbol{\phi}(k)\epsilon(k) \end{aligned} \quad (2.6)$$

As N tends to infinity, the estimated parameters tend to the true parameters, the second term on the RHS of Equation 2.6 vanishes. Denoting

$$\mathbf{R}_{\phi} \equiv \frac{1}{N}\sum_{k=1}^{N}\left(\boldsymbol{\phi}(k)\boldsymbol{\phi}^{T}(k)\right) \in \Re^{n\times n}$$

Equation 2.6 can be rewritten as

$$\boldsymbol{\xi}_{N}(\boldsymbol{\theta}) = \mathbf{R}_{\phi}\Delta\boldsymbol{\theta} + \frac{1}{\sqrt{N}}\sum_{k=1}^{N}\boldsymbol{\phi}(k)\epsilon(k)$$

Under the assumption that $\epsilon(k)$ is a Gaussian distributed white noise, it can be deduced that

$$\boldsymbol{\xi}_{N}(\boldsymbol{\theta}) \sim \aleph\left(\mathbf{R}_{\phi}\Delta\boldsymbol{\theta}, \mathbf{R}_{\phi}\sigma^{2}\right)$$

Hence, detecting any parametric change in the model is equivalent to checking whether $\boldsymbol{\xi}_{N}$ is zero mean. The occurrence of faults make the associated elements of the mean vector non zero. Therefore, fault isolation is reduced to finding out the non-zero elements in the mean vector.

2.7 Summary

A brief review of various multivariate statistical analysis approaches, e.g., PCA and PLS have been presented. Also few analytical redundancy based approaches for FDI have been discussed briefly. Dynamic PCA when applied to monitor a slowly time varying system triggers a number of false alarms. It highlights the need to have a recursive technique which can adapt to such small variations in the processes.

Chapter 3

Adaptive process monitoring via multi channel Lattice Filter

3.1 Introduction

Lattice filters were originally proposed by Itakura and Saito (1971) for speech analysis and synthesis. Since then lattice filters have been widely applied to adaptive signal processing, joint process estimation, e.g., fast startup equalizers and noise cancelling, adaptive control and system identification (Honig and Messerschmitt 1981, Lee *et al.* 1981, Friedlander 1983, Lev-Ari *et al.* 1984, Ljung 1987, Jabbari and Gibson 1988) for a number of reasons. The main advantage of the lattice filter is that it is recursive not only in time but also in order. Also, it is numerically efficient requiring a finite number $O(n)$ operations per time-update, where n is the order of the lattice filter algorithm.

As originally proposed (Itakura and Saito 1971), lattice filter can only be used for the identification of AR models. In the following years, the lattice filter has been extended for the identification of multivariate ARMA models (Lee *et al.* 1982, Jabbari and Gibson 1988, Kummert *et al.* 1992). Lee et al., (1982) depended on the embedding technique with the input and output processes being treated as a joint AR process. Jabbari and Gibson (1988) developed the vector channel lattice filter where the input signals are included in some channels in a specific manner. Kummeret et al., (1992) introduced a unified signal for the representation of input and output signals so that the ARMA model parameters can be estimated by applying the least squares criterion to the unified signals. The lattice filter extended least squares algorithm for the identification of scalar ARMAX model has been proposed by Friedlander (1983). However, as most industrial processes, e.g., chemical processes are multivariate in nature it is imperative to use vector ARMAX models to describe the dynamics of such processes. Also, Friedlander's algorithm is based on the assumption that input signals are noise free which is not applicable to the EIV based state space model case.

The EIV based state space model is the most generalized representation of any dynamic system. The benefit of using such a model is that the measurement noise in both the inputs

and outputs and the process noise can be taken into account. The EIV based state space model can be shown to be equivalent to a vector ARMAX model. The multi-channel lattice filter can be used for the identification of the vector ARMAX model and for the generation of residuals for process monitoring. Any change in the parameters or the structure of the process model is reflected as a change in the mean of the residual vector. As many industrial processes are fully dynamic and are often time varying, the residuals generated need to be updated recursively in *time* as well as in *order* to reduce the number of false alarms. The process monitoring is equivalent to checking if the mean of the recursively updated residual vector deviates from zero. By making use of the multi-channel lattice filter, residuals can be generated recursively in time and in order from the sampled data without estimating the model parameters explicitly. Multi-channel lattice filters can thus be used for adaptive process monitoring of slowly varying dynamic systems.

In the following sections the lattice filter algorithm, for the identification of vector ARMAX model and for the update of the residual vector in time and order simultaneously, is proposed. A unified signal is used to represent the input and output signals and accordingly a unified signal subspace is defined. The lattice filter algorithm has been derived from a geometric point of view by performing non-symmetric projection operations in the unified signal subspace. The computed residual vector is used for constructing the Hotelling T^2 statistic which is used as the monitoring index. The proposed approach is evaluated by monitoring a simulated process and a time-varying pilot plant.

The subject matter of this chapter is outlined as follows. The chapter starts with a brief introduction to lattice filter in Section 2. Lattice filter is discussed as it was originally proposed, i.e., for the identification of AR models. Section 3 outlines the problem formulation. Equivalence between the EIV based state space model and multi-channel lattice filter with respect to the generation of residual vector is presented in Section 4. Section 5 deals with the complete lattice filter algorithm for the recursive update of the residual vector in time and order. The order determination of the lattice filter and the computation of the T^2 statistic for process monitoring is considered in Section 6 which is followed by two case studies for the evaluation of the proposed algorithm in Section 7. The chapter ends with concluding remarks in Section 8.

3.2 Lattice Filter

The lattice filter is simply a result of performing a Gram Schmidt orthogonalization procedure on the data samples entering into the prediction. This section provides a brief overview of lattice filters for the identification of scalar AR models (Honig and Messerschmitt 1984). A geometric approach is taken for the derivation of the algorithms. The geometric approach lends an intuitive interpretation of many of the results which would otherwise seem to be merely mathematical manipulations. The section starts with the review of some mathematical preliminaries which are essential to the understanding of the later geometric

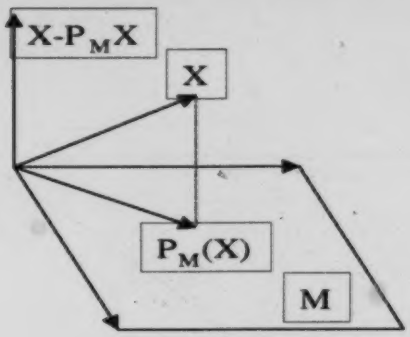


Figure 3.1: Illustration of projection theorem

derivations.

3.2.1 Mathematical Preliminaries

- A linear space consisting of infinite dimensional vectors is defined as

$$\mathbf{Y} \longleftrightarrow (\ldots y(-1), y(0), y(1), \ldots)$$

under the assumption that

$$\sum_{T} y^2(T) < \infty$$

which signifies that the total energy in the given data samples is finite.

- Inner product of two vectors is defined as

$$< \mathbf{X}, \mathbf{Y} > = \sum_{i=1}^{n} x_i y_i$$

- Hilbert Space - Linear space on which inner product is defined and has the property of completeness.

- Orthogonal Vectors - Two vectors are said to be orthogonal if their inner product is a zero (scalar), i.e.,

$$< \mathbf{X}, \mathbf{Y} > = 0$$

- Subspace - Subset of a linear space which itself is a linear space.

- Two subspaces M_1 and M_2 are said to be *orthogonal subspaces* if every pair of vectors, one taken from M_1 and the other taken from M_2, are orthogonal.

- The *projection theorem* is the basis for the derivation of lattice filter structure later in the chapter. It states that : Given a subspace M of a Hilbert space H and a vector

$\mathbf{X}$ in H there is a unique vector $\mathbf{P}_M\mathbf{X}$ in M called the *projection of* $\mathbf{X}$ *on* M which has the property that

$$< \mathbf{X} - \mathbf{P}_M\mathbf{X}, \mathbf{Y} > = 0$$

for every vector $\mathbf{Y}$ in $\mathbf{M}$. The concept of projection is illustrated in Figure 3.1 where the subspace M is the plane formed by the x-axis and y-axis and $\mathbf{X}$ is an arbitrary vector. A consequence of the projection theorem is that the projection $\mathbf{P}_M\mathbf{X}$ is the unique vector in M which is closest to $\mathbf{X}$, i.e.,

$$\| \mathbf{X} - \mathbf{P}_M\mathbf{X} \| < \| \mathbf{X} - \mathbf{Y} \|$$

for every $\mathbf{Y} \neq \mathbf{P}_M\mathbf{X}$ in M.

- Given two orthogonal subspaces M_1 and M_2 of Hilbert space H and an arbitrary vector $\mathbf{X}$ in H, the projection of $\mathbf{X}$ on $M_1 \oplus M_2$ can be expressed uniquely as

$$\mathbf{P}_{M_1 \oplus M_2}\mathbf{X} = \mathbf{P}_{M_1}\mathbf{X} + \mathbf{P}_{M_2}\mathbf{X}$$

or in words the sum of the projection on M_1 and the projection on M_2.

- Recalling the definition for the signal vector, the following notation is defined for the new vector obtained by delaying the signals by i samples,

$$z^{-i}\mathbf{Y} \longleftrightarrow (\ldots, y(-1-i), y(-i), y(1-i), \ldots).$$

In this notation, $z^{-i}\mathbf{Y}$ is a new vector obtained by delaying the samples in the $\mathbf{Y}$ vector by i samples.

- The prediction error of the n-th order, also termed as forward predictor error, is defined as follows

$$e_f(T \mid n) = y(T) - \sum_{j=1}^{n} f_j(n) y(T-j)$$

where a fixed filter coefficient vector $\mathbf{f}(n)$ has been assumed. The above in a vector form can be rewritten as

$$\mathbf{E}_f(n) = \mathbf{Y} - \sum_{j=1}^{n} f_j(n) z^{-j}\mathbf{Y}. \tag{3.1}$$

In a prediction problem, given a signal vector $\mathbf{Y}$, the objective is to minimize the length of the n-th order prediction error vector. The predictor is a vector in the subspace spanned by $z^{-1}\mathbf{Y}, \cdots, z^{-n}\mathbf{Y}$. This motivates us to define more generally the subspace spanned by $z^{-k}\mathbf{Y}, \cdots, z^{-m}\mathbf{Y}$, $k \leq m$, as $M(k,m)$. The summation term on the right hand side of Equation 3.1 (the forward predictor) is a general vector in $M(1,n)$, since it consists of a linear combination of the spanning vectors in $M(1,n)$. From the projection theorem, the

vector of the subspace $M(1,n)$ which is closest to $\mathbf{Y}$ is the projection of $\mathbf{Y}$ on $M(1,n)$, and thus

$$\mathbf{E}_f(n) = \mathbf{Y} - \mathbf{P}_{M(1,n)}\mathbf{Y} \tag{3.2}$$

is the error corresponding to the optimum predictor which has the property that

$$< \mathbf{E}_f(n), z^{-j}\mathbf{Y} > = 0,\ 1 \leq j \leq n$$

3.2.2 Derivation of Lattice filter

This section deals with the derivation of lattice filter structure using the mathematical tools of the previous section, and the projection theorem in particular.

The prediction problem at hand is to find the projection of $\mathbf{Y}$ on the subspace $M(1,n)$. The lattice filter is simply the consequence of finding a new set of vectors which also span the subspace $M(1,n)$, but which have the valuable property of being mutually orthogonal. A set of n vectors which satisfy this need is

$$\begin{aligned} \mathbf{E}_b(0) &= \mathbf{Y} \\ \mathbf{E}_b(m) &= z^{-m}\mathbf{Y} - \mathbf{P}_{M(0,m-1)}(z^{-m}\mathbf{Y}),\ 1 \leq m \leq n-1. \end{aligned} \tag{3.3}$$

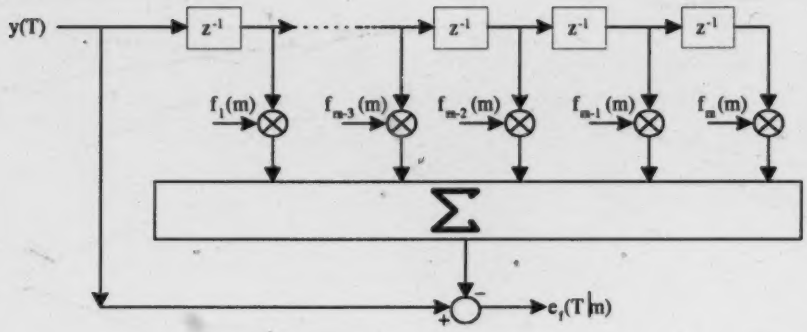
In particular, it will be shown later that $\mathbf{E}_b(m)$, $0 \leq m \leq n$, is an orthogonal basis for $M(0,n-1)$, or equivalently that $z^{-1}\mathbf{E}_b(m)$, $0 \leq m \leq n$ is an orthogonal basis for $M(1,n)$. Each of the vectors in Equation 3.3 can be written in the form

$$\mathbf{E}_b(m) = z^{-m}\mathbf{Y} - \sum_{j=1}^{m} b_j(m) z^{-j+1}\mathbf{Y} \tag{3.4}$$

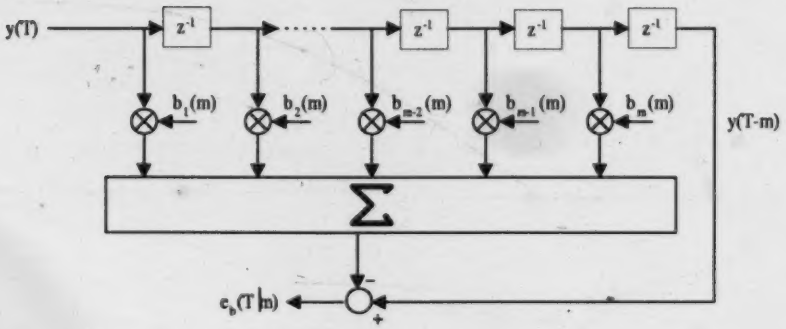
for some particular constants $b_j(m)$ corresponding to the projection in Equation 3.3. The time domain equivalent of the above equation is

$$e_b(T \mid m) = y(T-m) - \sum_{j=1}^{m} b_j(m) y(T-j+1). \tag{3.5}$$

Figure 3.2 illustrate the realization of m-th order forward and backward predictors. In Figure 3.2.1, $e_f(T \mid m)$ is the error in predicting a current sample in terms of a linear combination of m past samples. In Figure 3.2.2, $e_b(T \mid m)$ is the error in predicting a sample m sampling intervals in the past in terms of a linear combination of m more recent samples, up to and including the current sample. For this reason, the predictor of Equation 3.4 is called a *backward predictor*; $e_b(T \mid m)$ is called the m-th order backward prediction error; and the $b_j(m)$ are called the backward prediction coefficients. The vectors in Equation 3.2 are simply the consequence of applying *Gram-Schmidt orthogonalization procedure* to the vectors $\mathbf{Y}$,$z^{-1}\mathbf{Y}$,...,$z^{-n+1}\mathbf{Y}$. These different orders of backward predictors form an orthogonal basis for $M(0,n-1)$ as explained below.



3.2.1: Forward predictor



3.2.2: Backward predictor

Figure 3.2: Realization of m-th order predictors

- The successive orders of backward prediction error span $M(0, n-1)$. It can be explained by the fact that $\mathbf{E}_b(m)$ contains as one component $z^{-m}\mathbf{Y}$, and hence each of $\mathbf{Y}, z^{-1}\mathbf{Y}, \ldots, z^{-n+1}\mathbf{Y}$ is included in one or more of the backward prediction errors.

- The backward prediction errors are mutually orthogonal. By definition of the projection, $\mathbf{E}_b(m)$ is orthogonal to $M(0, n-1)$. But $\mathbf{E}_b(0)$ through $\mathbf{E}_b(m-1)$ are vectors in the subspace $M(0, n-1)$, since they can be written as linear combinations of $\mathbf{Y}, \ldots, z^{-m+1}\mathbf{Y}$. Since each $\mathbf{E}_b(m)$ is orthogonal to $\mathbf{E}_b(0)$ through $\mathbf{E}_b(m-1)$, it follows that the different orders of prediction error are mutually orthogonal.

The backward predictors have no important direct physical significance, except as an intermediate signal in the construction of the lattice filter. It can be seen later that the lattice filter realizing the n-th order forward predictor generates all lower orders of backward prediction error. Also the backward and forward prediction errors have the same norm, i.e. they have the same length,

$$\|\mathbf{E}_f(m)\|^2 = \|\mathbf{E}_b(m)\|^2, \ 1 \leq m \leq n.$$

The lattice structure is defined by the equations

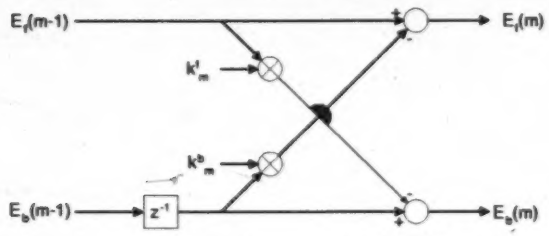
$$\begin{aligned} \mathbf{E}_f(m) &= \mathbf{E}_f(m-1) - k_m^b z^{-1}\mathbf{E}_b(m-1), \ \mathbf{E}_f(0) = \mathbf{Y} \\ \mathbf{E}_b(m) &= z^{-1}\mathbf{E}_b(m-1) - k_m^f \mathbf{E}_f(m-1), \ \mathbf{E}_b(0) = \mathbf{Y} \end{aligned}$$

for $1 \leq m \leq n$ where n is the order of the desired forward predictor and k_m^b and k_m^f are appropriate constants. These are called *order update equations* since they relate higher order forward and backward prediction errors to lower order prediction errors. In the time domain, the order updates become

$$\begin{aligned} e_f(T \mid m) &= e_f(T \mid m-1) - k_m^b e_b(T-1 \mid m-1) \\ e_f(T \mid 0) &= y(T) \\ e_b(T \mid m) &= e_b(T-1 \mid m-1) - k_m^f e_f(T \mid m-1) \\ e_b(T \mid 0) &= y(T). \end{aligned}$$

The lattice filter structure is shown in Figure 3.3. In Figure 3.3.1, the successive stages of the filter develop the successive higher order forward and backward prediction errors. In finally developing an n-th order forward predictor at the final output, all lower order prediction errors are developed as intermediate signals between the stages. A single stage of the lattice filter is shown in Figure 3.3.2. The inputs to a lattice filter stage are the forward and backward prediction errors from the $(m-1)$-th stage and outputs are the forward and backward prediction errors using an m-th order filter. The inputs to the first lattice stage are $\mathbf{Y}$, which also corresponds to both prediction errors of order 0.

3.3.1: Overall structure



3.3.2: Single stage

Figure 3.3: Lattice filter based n-th order predictor

The coefficients k_m^b and k_m^f are called *partial correlation* or *PARCOR* coefficients. The *PARCOR* coefficients are independent of the filter order n. Successive lattice stages may therefore be added or existing stages subtracted without the necessity of recalculating already existing *PARCOR* coefficients. The fact that lower orders of prediction error are developed in realizing an n-th order filter can be used when the order of the predictor is not known in advance. Successive stages can be added and order determination techniques can be applied on the respective error vectors for the determination of the order. Once the order has been determined, additional stages can be removed without affecting the lower stages of the filter.

3.3 Problem Formulation

Having presented a conceptual and geometric introduction to the lattice filter, the following sections deal with the identification of vector ARMAX model using multi channel lattice filter. This section starts with the problem formulation for the identification purpose.

Assume that the normal behaviour of the process under consideration be represented by the unknown discrete state space model, as follows: (Chou and Verhaegen 1997)

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\tilde{\mathbf{u}}(k) + \mathbf{K}\mathbf{p}(k) \\ \tilde{\mathbf{y}}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{D}\tilde{\mathbf{u}}(k) \end{aligned} \tag{3.6}$$

where $\tilde{\mathbf{u}}(k) \in \Re^l$ and $\tilde{\mathbf{y}}(k) \in \Re^m$ are noise-free inputs and outputs, respectively, and $\mathbf{x}(k) \in \Re^n$ is the state vector. In addition, $\mathbf{p}(k) \in \Re^d$ is the process noise, and $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ and $\mathbf{K}$ are system and noise gain matrices with appropriate dimensions respectively.

It is assumed that the observed input $\mathbf{u}(k)$ and output $\mathbf{y}(k)$ are corrupted by zero-mean Gaussian distributed noise vectors $\mathbf{v}(k)$ and $\mathbf{o}(k)$, respectively, i.e.

$$\begin{array}{rcl} \mathbf{u}(k) & = & \tilde{\mathbf{u}}(k) + \mathbf{v}(k) \\ \mathbf{y}(k) & = & \tilde{\mathbf{y}}(k) + \mathbf{o}(k) \end{array} \tag{3.7}$$

Substituting Equation 3.7 into Equation 3.6 and performing a straightforward algebraic manipulation, the following equation is obtained

$$\mathbf{y}_n(k) = \mathbf{\Gamma}_n \mathbf{x}(k-n) + \mathbf{H}_n \mathbf{u}_n(k) - \mathbf{H}_n \mathbf{v}_n(k) + \mathbf{G}_n \mathbf{p}_n(k) + \mathbf{o}_n(k) \tag{3.8}$$

where

$$\mathbf{y}_n(k) = \begin{bmatrix} \mathbf{y}(k) \\ \vdots \\ \mathbf{y}(k-n) \end{bmatrix} \in \Re^{m_n}$$

is the augmented output vector,

$$\mathbf{\Gamma}_n = \begin{bmatrix} \mathbf{CA}^n \\ \vdots \\ \mathbf{C} \end{bmatrix} \in \Re^{m_n \times n}$$

is the extended observability matrix with rank n,

$$\mathbf{H}_n = \begin{bmatrix} \mathbf{D} & \mathbf{CB} & \cdots & \cdots & \mathbf{CA}^{n-1}\mathbf{B} \\ \mathbf{0} & \mathbf{D} & \mathbf{CB} & \cdots & \vdots \\ \vdots & & \ddots & \ddots & \\ & & & \mathbf{D} & \mathbf{CB} \\ \mathbf{0} & \cdots & & & \mathbf{D} \end{bmatrix} \in \Re^{m_n \times l_n}$$

and

$$\mathbf{G}_n = \begin{bmatrix} \mathbf{0} & \mathbf{CK} & \cdots & \cdots & \mathbf{CA}^{n-1}\mathbf{K} \\ & \mathbf{0} & \mathbf{CK} & \cdots & \vdots \\ \vdots & & \ddots & \ddots & \\ & & & \mathbf{0} & \mathbf{CK} \\ \mathbf{0} & \cdots & & & \mathbf{0} \end{bmatrix} \in \Re^{m_n \times d_n}$$

are two upper triangular block Toeplitz matrices with $m_n = (n+1)m$, $l_n = (n+1)l$ and $d_n = (n+1)d$. The vectors $\mathbf{v}_n(k) \in \Re^{l_n}$, $\mathbf{o}_n(k) \in \Re^{m_n}$, $\mathbf{p}_n(k) \in \Re^{d_n}$, and $\mathbf{u}_n(k) \in \Re^{l_n}$ are defined in the same format as $\mathbf{y}_n(k)$. Assuming that the characteristic polynomial of $\mathbf{A} - \lambda\mathbf{I}$ is as follows:

$$\det(\mathbf{A} - \lambda\mathbf{I}) = \lambda^n + \alpha_1\lambda^{n-1} +, \cdots, +\alpha_{n-1}\lambda + \alpha_n = 0$$

According to the Cayley-Hamilton theorem, pre-multiplying both sides of Equation 3.8 by the following matrix

$$\left(\mathbf{\Gamma}_n^{\perp}\right)^T = [\mathbf{I}_m \ \alpha_1\mathbf{I}_m \cdots \alpha_n\mathbf{I}_m \,] \in \Re^{m \times m_n}$$

leads to

$$\mathbf{y}(k)+\alpha_1\mathbf{y}(k-1)+,\cdots,+\alpha_{n-1}\mathbf{y}(k-n+1)+\alpha_n\mathbf{y}(k-n)=$$
$$\left(\Gamma_n^{\perp}\right)^T\mathbf{H}_n\mathbf{u}_n(k)+\left(\Gamma_n^{\perp}\right)^T\mathbf{o}_n(k)-\left(\Gamma_n^{\perp}\right)^T\mathbf{H}_n\mathbf{v}_n(k)+\left(\Gamma_n^{\perp}\right)^T\mathbf{G}_n\mathbf{p}_n(k) \qquad (3.9)$$

where $\Gamma_n^{\perp}$ is a subspace of the orthogonal complement of Γ_n.

According to the model definition given by Ljung (1987), Equation 3.9 is a typical ARMAX process. Denoting

$$\begin{aligned}\left(\Gamma_n^{\perp}\right)^T\mathbf{H}_n &= \left[\mathbf{D}\ \ \mathbf{CB}+\alpha_1\mathbf{D}\ \ \mathbf{CAB}+\alpha_1\mathbf{CB}+\alpha_2\mathbf{D}\cdots\sum_{j=0}^{n-1}\alpha_j\mathbf{CA}^{n-1-j}\mathbf{B}+\alpha_n\mathbf{D}\right]\\ &\equiv \left[\mathbf{L}^{(0)}\ \mathbf{L}^{(1)}\ \cdots\mathbf{L}^{(n)}\right]\in\Re^{m\times l_n}\end{aligned}$$

and

$$\begin{aligned}\left(\Gamma_n^{\perp}\right)^T\mathbf{G}_n &= \left[\mathbf{0}\ \ \mathbf{CK}\ \ \mathbf{CAK}+\alpha_1\mathbf{CK}\ \ \cdots\sum_{j=0}^{n-1}\alpha_j\mathbf{CA}^{n-1-j}\mathbf{K}\right]\\ &\equiv \left[\mathbf{0}\ \ \mathbf{\Psi}^{(1)}\ \cdots\mathbf{\Psi}^{(n)}\right]\in\Re^{m\times l_n}\end{aligned}$$

Equation 3.9 can be rewritten as

$$\begin{aligned}\mathbf{e}_n(k) &\equiv \mathbf{y}(k)+\sum_{s=1}^{n}\alpha_s\mathbf{y}(k-s)-\sum_{\omega=0}^{n}\mathbf{L}^{(\omega)}\mathbf{u}(k-\omega)\\ &= \mathbf{o}(k)+\sum_{s=1}^{n}\alpha_s\mathbf{o}(k-s)-\sum_{s=0}^{n}\mathbf{L}^{(s)}\mathbf{v}(k-s)+\sum_{s=1}^{n}\mathbf{\Psi}^{(s)}\mathbf{p}(k-s) \qquad (3.10)\end{aligned}$$

where $\mathbf{e}_n(k)$ is defined as the residual vector and will be used for process monitoring. The first line on the right hand side (RHS) in Equation 3.10 is the computational form of $\mathbf{e}_n(k)$, showing how $\mathbf{e}_n(k)$ is computed based on the sampled data $\{\mathbf{y}(k)\ \ \mathbf{u}(k)\}$ and their time-lagged values $\{\mathbf{y}(k-1)\ \ \mathbf{u}(k-1)\ \cdots\mathbf{y}(k-n)\ \ \mathbf{u}(k-n)\}$; the second line on the RHS is the internal form, showing how $\mathbf{e}_n(k)$ depends on the measurement noise $[\mathbf{o}(k)\ \ \mathbf{v}(k)]$ and the process noise $\mathbf{p}(k)$ and their time-lagged values.

Apparently, Equation 3.10 can be decomposed into m independent equations, e.g., the i^{th} residual equation for $i=1\cdots m$ is

$$\begin{aligned}e_n^i(k) &= y_i(k)-\left(\boldsymbol{\theta}_n^i\right)^T\boldsymbol{\xi}_n^i(k)\\ &= \sum_{s=0}^{n}\alpha_s o_i(k-s)-\sum_{s=0}^{n}\mathbf{l}^{(s)}(i,:)\mathbf{v}(k-s)+\sum_{s=1}^{n}\boldsymbol{\psi}^{(s)}(i,:)\mathbf{p}(k-s) \qquad (3.11)\end{aligned}$$

where $\alpha_0=1$; $y_i(k)$ and $o_i(k)$ are the i^{th} element of the vectors $\mathbf{y}(k)$ and $\mathbf{o}(k)$, respectively;

in addition,

$$\boldsymbol{\xi}_n^i(k) = \begin{bmatrix} \mathbf{u}(k) \\ y_i(k-1) \\ \mathbf{u}(k-1) \\ y_i(k-2) \\ \mathbf{u}(k-2) \\ \vdots \\ y_i(k-n) \\ \mathbf{u}(k-n) \end{bmatrix} \in \Re^{l_n+n} \tag{3.12}$$

and $\boldsymbol{\theta}_n^i = \left[\mathbf{l}^{(0)}(i,:) \ -\alpha_1 \ \mathbf{l}^{(1)}(i,:) \ -\alpha_2 \ \mathbf{l}^{(2)}(i,:) \cdots -\alpha_n \ \mathbf{l}^{(n)}(i,:)\right]^T \in \Re^{(l_n+n)}$, with $\mathbf{l}^{(\bullet)}(i,:)$ and $\boldsymbol{\psi}^{(\bullet)}(i,:)$ symbolizing the i^{th} row of the matrices $\mathbf{L}^{(\bullet)}$ and $\boldsymbol{\Psi}^{(\bullet)}$, respectively.

Using Equation 3.11, Equation 3.10 can be rewritten into

$$\begin{aligned} \mathbf{e}_n(k) \ &\equiv \ \begin{bmatrix} e_n^1(k) \\ e_n^2(k) \\ \vdots \\ e_n^m(k) \end{bmatrix} = \mathbf{y}(k) - \begin{bmatrix} \left(\boldsymbol{\theta}_n^1\right)^T \boldsymbol{\xi}_n^1(k) \\ \left(\boldsymbol{\theta}_n^2\right)^T \boldsymbol{\xi}_n^2(k) \\ \vdots \\ \left(\boldsymbol{\theta}_n^m\right)^T \boldsymbol{\xi}_n^m(k) \end{bmatrix} \\ &= \ \mathbf{o}(k) - \mathbf{L}^{(0)}\mathbf{v}(k) + \sum_{s=1}^{n} \left(\alpha_s \mathbf{o}(k-s) - \mathbf{L}^{(s)}\mathbf{v}(k-s) + \boldsymbol{\Psi}^{(s)}\mathbf{p}(k-s)\right) \end{aligned} \tag{3.13}$$

The following remarks are in order, on the basis of the examination of the above residual vector:

1. $\mathbf{e}_n(k)$ consists of m elements $\{e_n^1(k) \cdots e_n^m(k)\}$. To generate $\mathbf{e}_n(k)$, an m-channel lattice filter is needed. However, since each element of $\mathbf{e}_n(k)$ is independent of the other, in practice the algorithm of the lattice filter has to be derived for the i^{th} channel with $i \in [1, \cdots, m]$.

2. The model parameters of $\mathbf{e}_n(k)$ are

$$\boldsymbol{\Theta}_n \equiv \begin{bmatrix} \boldsymbol{\theta}_n^1 \\ \boldsymbol{\theta}_n^2 \\ \vdots \\ \boldsymbol{\theta}_n^m \end{bmatrix} \in \Re^{m(l_n+n)}$$

   However, it is proposed to generate $\mathbf{e}_n(k)$ recursively in time and order by using the sampled data $\left[\mathbf{y}_n^T(k) \ \mathbf{u}_n^T(k)\right]^T$ without estimating the model parameters explicitly.

3. $\mathbf{e}_n(k)$ will be a vector moving average (MA) process with the zero mean noise vector $[\mathbf{o}(k) \ \ \mathbf{v}(k) \ \ \mathbf{p}(k)]$ as input and its covariance matrix is

$$\begin{aligned} \mathbf{R}_n^{\mathbf{e}}(k) \ &= \ \sum_{s=0}^{n} \alpha_s^2 \mathrm{E}\{\mathbf{o}(k)\,(\mathbf{o}(k))^T\} + \\ &\sum_{s=0}^{n} \mathbf{L}^{(s)} \mathrm{E}\{\mathbf{v}(k)\,(\mathbf{v}(k))^T\} \left(\mathbf{L}^{(s)}\right)^T + \sum_{s=1}^{n} \boldsymbol{\Psi}^{(s)} \mathrm{E}\{\mathbf{p}(k)\,(\mathbf{p}(k))^T\} \left(\boldsymbol{\Psi}^{(s)}\right)^T \in \Re^{m \times m} \end{aligned}$$

4. The residuals, properly normalized, as in the local approach (see Section 2.6) will indicate any change in process parameters or the process structure.

Therefore, process monitoring is equivalent to checking if the mean of the recursively updated $\mathbf{e}_n(k)$ deviates from zero.

3.4 Residual generation using multi-channel Lattice filter

In this section, the residual vector $\mathbf{e}_n(k)$ is generated, by the use of the multi-channel lattice filter, directly from the sampled data. The following section starts with the introduction of a unified signal and the associated subspace of the signal.

3.4.1 Notations and definitions

It follows directly from Equation 3.11 that $e_n^i(k)$ is a linear combination of following signals:

$$\{y_i(k)\ u_1(k)\ \cdots\ u_l(k)\ y_i(k-1)\ u_1(k-1)\ \cdots\ u_l(k-1)\cdots\ y_i(k-n)\ u_1(k-n)\ \cdots\ u_l(k-n)\}$$

where, $u_j(k)$ for $j = 1 \cdots l$ also stands for the j^{th} element of the input vector $\mathbf{u}(k)$.

The key step for the multi-channel lattice filter to generate the residual $e_n^i(k)$ is to introduce a unified notation to represent all the input and output signals so that they can be handled in a unified frame work. The unified signal (Kummert et al., 1992) is defined as

$$z^i(kM-s) = \begin{cases} y_i(k), & s = 1. \\ u_{s-1}(k), & s = 2 \cdots M, \text{ where } M = l+1. \end{cases} \tag{3.14}$$

With Equation 3.14, each of the aforementioned signals has a one-to-one unique correspondence to the unified signal $z^i(tM-s)$ for $\{s = 1 \cdots M,\ t = k \cdots k-n\}$, e.g.,

$$\begin{bmatrix} z^i(kM-1) \\ z^i(kM-2) \\ z^i(kM-3) \\ \vdots \\ z^i(kM-M) \\ z^i((k-1)M-1) \\ z^i((k-1)M-2) \\ z^i((k-1)M-3) \\ \vdots \\ z^i((k-1)M-M) \\ \vdots \\ z^i((k-n)M-1) \\ z^i((k-n)M-2) \\ z^i((k-n)M-3) \\ \vdots \\ z^i((k-n)M-M) \end{bmatrix} \equiv \begin{bmatrix} y_i(k) \\ u_1(k) \\ u_2(k) \\ \vdots \\ u_l(k) \\ y_i(k-1) \\ u_1(k-1) \\ u_2(k-1) \\ \vdots \\ u_l(k-1) \\ \vdots \\ y_i(k-n) \\ u_1(k-n) \\ u_2(k-n) \\ \vdots \\ u_l(k-n) \end{bmatrix} \tag{3.15}$$

A signal vector with infinite length is defined as

$$\mathbf{z}^i_{kM-s} = \left[z^i(kM-s)\ z^i((k-1)M-s)\cdots z^i(1M-s)\ 0\cdots 0\right]^T$$

which belongs to the Hilbert space, e.g., the vector $\mathbf{z}^i_{kM-s}$ has real elements only, and for $\mathbf{z}^i_{kM-s}$ and $\mathbf{z}^i_{(k-\mu)M-s}$, their inner product is

$$< \mathbf{z}^i_{kM-s}, \mathbf{z}^i_{(k-\mu)M-s} >= \sum_{t=1}^{k} \lambda^{k-t} z^i(tM-s) z^i((t-\mu)M-s) < \infty$$

where $s = 1\cdots M$, $z^i(t) = 0$ for $t < 0$, and $0 < \lambda \leq 1$ is a forgetting factor.

Again defining a subspace as

$$\mathbf{h}^i_{s,\eta}(k) = \text{Span}\{\mathbf{z}^i_{kM-s-1}, \mathbf{z}^i_{kM-s-2}, \cdots, \mathbf{z}^i_{kM-s-\eta}\} \tag{3.16}$$

which is spanned by columns of the following matrix

$$\mathbf{z}^i_{kM-s-1:kM-s-\eta} = \left[\mathbf{z}^i_{kM-s-1}\ \mathbf{z}^i_{kM-s-2}\ \cdots\ \mathbf{z}^i_{kM-s-\eta}\right] \in \Re^{\infty\times\eta},\quad \eta = 1\cdots M_n - 1. \tag{3.17}$$

where $M_n = M(n+1)$. Further, defining

$$\mathbf{f}^i_{s,\eta}(k) = \mathbf{z}^i_{kM-s} - \mathbf{z}^i_{kM-s}|_{\mathbf{h}^i_{s,\eta}(k)},\quad \eta = 1\cdots M_n - 1;\ s = 1\cdots M. \tag{3.18}$$

as the forward error vector of the IV (Instrument Variable) lattice filter, which is the difference between the vector $\mathbf{z}^i_{kM-s}$ and its non-symmetric projection onto the subspace $\mathbf{h}^i_{s,\eta}(k)$. As will be shown later, while the conventional least squares lattice filter can be derived by performing a symmetric projection onto the subspace $\mathbf{h}^i_{s,\eta}(k)$, to derive the IV lattice filter, a non-symmetric projection on the subspace has to be performed.

The non-symmetric projection of $\mathbf{z}^i_{kM-s}$ onto the subspace $\mathbf{h}^i_{s,\eta}(k)$ is

$$\begin{aligned}\mathbf{z}^i_{kM-s}|_{\mathbf{h}^i_{s,\eta}(k)} \;=\; & \mathbf{z}^i_{kM-s-1:kM-s-\eta}\left(\left(\mathbf{z}^i_{(k-\mu)M-s-1:(k-\mu)M-s-\eta}\right)^T \mathbf{z}^i_{kM-s-1:kM-s-\eta}\right)^{-1} \cdot \\ & \left(\mathbf{z}^i_{(k-\mu)M-s-1:(k-\mu)M-s-\eta}\right)^T \mathbf{z}^i_{kM-s}\end{aligned} \tag{3.19}$$

Geometrically, $\mathbf{f}^i_{s,\eta}(k)$ is perpendicular to the subspace $\mathbf{h}^i_{s,\eta}(k-\mu)$ where μ is a positive integer. Note that the non-symmetric projection defined above is reduced to the conventional symmetric projection when $\mu = 0$.

Denote the first element of $\mathbf{f}^i_{s,\eta}(k)$ by

$$\varepsilon^i_{s,\eta}(k) \;\equiv\; \boldsymbol{\phi}^T \mathbf{f}^i_{s,\eta}(k) \tag{3.20}$$

with $\boldsymbol{\phi} = [1\ 0\cdots 0]^T \in \Re^{\infty}$

In addition to the forward error vector $\mathbf{f}^i_{s,\eta}(k)$, the complete lattice filter algorithm also includes other quantities.

Define

$$\mathbf{b}_{s,\eta}^{i}(k) = \mathbf{z}_{kM-s-\eta}^{i} - \mathbf{z}_{kM-s-\eta}^{i}|_{\mathbf{h}_{s-1,\eta}^{i}(k)}, \quad \eta = 1 \cdots M_n - 1; \; s = 1 \cdots M. \tag{3.21}$$

as the backward error vector, which is the difference between the vector $\mathbf{z}_{kM-s-\eta}^{i}$ and its non-symmetric projection onto the subspace $\mathbf{h}_{s-1,\eta}^{i}(k)$. Geometrically, $\mathbf{b}_{s,\eta}^{i}(k)$ is perpendicular to the subspace $\mathbf{h}_{s-1,\eta}^{i}(k-\mu)$.

Similarly, denote the first element of $\mathbf{b}_{s,\eta}^{i}(k)$ by

$$\gamma_{s,\eta}^{i}(k) = \phi^{T}\mathbf{b}_{s,\eta}^{i}(k) \tag{3.22}$$

Furthermore, define

$$\phi_{\mathbf{h}_{s,\eta}^{i}(k)} = \phi - \phi|_{\mathbf{h}_{s,\eta}^{i}(k)} \tag{3.23}$$

which is orthogonal to the subspace $\mathbf{h}_{s,\eta}^{i}(k-\mu)$.

More general, for any two vectors $\mathbf{a}_k \in \Re^{\infty}$ and $\mathbf{b}_k \in \Re^{\infty}$ located in the afore-mentioned Hilbert subspace, their inner product is defined by

$$< \mathbf{a}_k, \mathbf{b}_k >= \mathbf{a}_k^{T}\mathbf{\Lambda}\mathbf{b}_k$$

where

$$\mathbf{\Lambda} = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & \lambda & 0 & \cdots & 0 \\ 0 & 0 & \lambda^2 & \cdots & \\ \vdots & & & \ddots & \vdots \\ 0 & \cdots & & & \lambda^{\infty} \end{bmatrix} \in \Re^{\infty \times \infty}$$

Also for any vector $\mathbf{a}_k \in \Re^{\infty}$ belonging to the afore-mentioned Hilbert space, its non-symmetric projection $\mathbf{a}_k|_{\mathbf{h}_\eta(k)}$ onto the following subspace

$$\mathbf{h}_\eta(k) = \text{Span}\{\mathbf{s}_{k-1}, \cdots, \mathbf{s}_{k-\eta}\} \in \Re^{\infty}$$

is computed as

$$\mathbf{a}_k|_{\mathbf{h}_\eta(k)} \equiv \mathbf{s}_{k-1:k-\eta}\left(\mathbf{s}_{k-\mu-1:k-\mu-\eta}^{T}\mathbf{\Lambda}\mathbf{s}_{k-1:k-\eta}\right)^{-1}\mathbf{s}_{k-\mu-1:k-\mu-\eta}^{T}\mathbf{\Lambda}\mathbf{a}_k \tag{3.24}$$

where

$$\mathbf{s}_{k-1:k-\eta} = [\mathbf{s}_{k-1} \cdots \mathbf{s}_{k-\eta}]$$

consists of vectors located in the Hilbert space. Further, the difference between $\mathbf{a}_k$ and $\mathbf{a}_k|_{\mathbf{h}_\eta(k)}$ is defined as the projection error, e.g.,

$$(\mathbf{a}_k)_{\mathbf{h}_\eta(k)} \equiv \mathbf{a}_k - \mathbf{a}_k|_{\mathbf{h}_\eta(k)} \tag{3.25}$$

Finally, there is a need to define several scalar quantities, which are inner products of the associated vectors,

$$\alpha_{s,\eta}^{i}(k) =< \mathbf{f}_{s,\eta}^{i}(k-\mu), \mathbf{f}_{s,\eta}^{i}(k) >, \; s = 1 \cdots M. \tag{3.26}$$

$$\beta^i_{s,\eta}(k) = < \mathbf{b}^i_{s,\eta}(k-\mu), \mathbf{b}^i_{s,\eta}(k) >,\ s = 1 \cdots M. \tag{3.27}$$

$$\epsilon^{i,\mu}_{s,\eta}(k) = < \mathbf{f}^i_{s,\eta}(k-\mu), \phi_{\mathbf{h}^i_{s,\eta}(k)} >,\ s = 1 \cdots M. \tag{3.28}$$

$$\varsigma^{i,\mu}_{s,\eta}(k) = \begin{cases} < \mathbf{b}^i_{s+1,\eta}(k-\mu), \phi_{\mathbf{h}^i_{s,\eta}(k)} >, & s = 1 \cdots M-1. \\ < \mathbf{b}^i_{1,\eta}(k-1-\mu), \phi_{\mathbf{h}^i_{s,\eta}(k)} >, & s = M. \end{cases} \tag{3.29}$$

$$\pi^i_{s,\eta+1}(k) = 1 - < \phi_{\mathbf{h}^i_{s-1,\eta+1}(k-\mu)}, \phi_{\mathbf{h}^i_{s-1,\eta+1}(k)} >,\ s = 1 \cdots M. \tag{3.30}$$

$$\varpi^i_{s,\eta}(k) = \begin{cases} < \mathbf{f}^i_{s,\eta-1}(k-\mu), \mathbf{b}^i_{s+1,\eta-1}(k) >, & s = 1 \cdots M-1. \\ < \mathbf{f}^i_{s,\eta-1}(k-\mu), \mathbf{b}^i_{1,\eta-1}(k-1) >, & s = M. \end{cases} \tag{3.31}$$

and

$$\psi^i_{s,\eta}(k) \equiv \begin{cases} < \mathbf{f}^i_{s,\eta-1}(k), \mathbf{b}^i_{s+1,\eta-1}(k-\mu) >, & s = 1 \cdots M-1. \\ < \mathbf{f}^i_{s,\eta-1}(k), \mathbf{b}^i_{1,\eta-1}(k-1-\mu) >, & s = M. \end{cases} \tag{3.32}$$

Note that for $\mu > 0$, generally (Box and Jenkins, 1976),

$$\varpi^i_{s,\eta}(k) \neq \psi^i_{s,\eta}(k)$$

3.4.2 Associating the Lattice Filter with Parameters $\theta^i_n(k)$

After introducing the notations and definitions for lattice filter, in this subsection, lattice filter is associated with the parameters $\mathbf{e}_n(k)$.

From Equations 3.18, 3.20, 3.24 and 3.25, it follows

$$\begin{aligned} \varepsilon^i_{1,M(n+1)-1}(k) &= z^i(kM-1) - \left(\mathbf{z}^i_{kM-1}\right)^T \mathbf{\Lambda} \mathbf{z}^i_{(k-\mu)M-2:(k-\mu)M-M_n} \bullet \\ &\quad \left(\left(\mathbf{z}^i_{kM-2:kM-M_n}\right)^T \mathbf{\Lambda} \mathbf{z}^i_{(k-\mu)M-2:(k-\mu)M-M_n}\right)^{-1} \begin{bmatrix} z^i(kM-2) \\ \vdots \\ z^i(kM-M_n) \end{bmatrix} \end{aligned}$$

On the other hand, it follows from Equations 3.12 and 3.14 that

$$z^i(kM-1) = y_i(k)$$

and

$$\boldsymbol{\xi}^i_n(k) = \begin{bmatrix} z^i(kM-2) \\ \vdots \\ z^i(kM-M_n) \end{bmatrix}$$

then it can be inferred that

$$\begin{aligned} \varepsilon^i_{1,M(n+1)-1}(k) &= y_i(k) - \left(\mathbf{z}^i_{kM-1}\right)^T \mathbf{\Lambda} \mathbf{z}^i_{(k-\mu)M-2:(k-\mu)M-M_n} \bullet \\ &\quad \left(\left(\mathbf{z}^i_{kM-2:kM-M_n}\right)^T \mathbf{\Lambda} \mathbf{z}^i_{(k-\mu)M-2:(k-\mu)M-M_n}\right)^{-1} \boldsymbol{\xi}^i_n(k) \end{aligned} \tag{3.33}$$

Moreover, from Equation 3.11, it can be concluded that for $i = 1 \cdots m$,

$$\varepsilon^i_{1,M(n+1)-1}(k) \equiv e^i_n(k) \tag{3.34}$$

and

$$\boldsymbol{\theta}^i_n \equiv \left(\left(\mathbf{z}^i_{(k-\mu)M-2:(k-\mu)M-M_n} \right)^T \mathbf{\Lambda} \mathbf{z}^i_{kM-2:kM-M_n} \right)^{-1} \left(\mathbf{z}^i_{(k-\mu)M-2:(k-\mu)M-M_n} \right)^T \mathbf{\Lambda} \mathbf{z}^i_{kM} \tag{3.35}$$

Equations 3.34 and 3.35 establish the link between the i^{th} residual $e^i_n(k)$ of the system and the lattice filter, making the generation of $e^i_n(k)$ possible by the use of the lattice filter without identifying the parameters $\boldsymbol{\theta}^i_n$ explicitly.

3.5 Lattice filter Algorithm

This section is the core of the chapter, where the complete IV lattice filter algorithm including the order-recursion, time-recursion, and parameter recursion equations are developed. The section starts with associating the IV lattice filter with the ARMAX model parameters $\boldsymbol{\theta}^i_n(k)$.

3.5.1 Order Recursion Equations

The complete IV lattice filter contains three parts, in which the order-recursion equations must be derived first.

Using Equations 3.16 and 3.21 results in the following decomposition:

$$\mathbf{h}^i_{s,\eta}(k) = \mathbf{h}^i_{s,\eta-1}(k) \oplus \text{Span}\{\mathbf{b}^i_{s+1,\eta-1}(k)\} \tag{3.36}$$

where the symbol $\oplus$ stands for the sum of two subspaces. Subsequently, in terms of the projection formula given by Equation 3.24, it can be shown that

$$\begin{aligned} \mathbf{f}^i_{s,\eta}(k) &= \mathbf{z}^i_{kM-s} - \mathbf{z}^i_{kM-s}|_{\mathbf{h}^i_{s,\eta-1}(k) \oplus \text{Span}\{\mathbf{b}^i_{s+1,\eta-1}(k)\}} \\ &= \mathbf{f}^i_{s,\eta-1}(k) - \frac{\psi^i_{s,\eta}(k)}{\beta^i_{s+1,\eta-1}(k)} \mathbf{b}^i_{s+1,\eta-1}(k), \quad s = 1 \cdots M-1. \end{aligned} \tag{3.37}$$

The detailed derivation of Equation 3.37 is given in Appendix A.1.

It follows from re-using Equations 3.16 and 3.18 that

$$\mathbf{h}^i_{s-1,\eta}(k) = \mathbf{f}^i_{s,\eta-1}(k) \oplus \mathbf{h}^i_{s,\eta-1}(k) \tag{3.38}$$

Then, following the similar steps to derive Equation 3.37,

$$\begin{aligned} \mathbf{b}^i_{s,\eta}(k) &= \mathbf{z}^i_{kM-s-\eta} - \mathbf{z}^i_{kM-s-\eta}|_{\mathbf{h}^i_{s,\eta-1}(k) \oplus \mathbf{f}^i_{s,\eta-1}(k)} \\ &= \mathbf{b}^i_{s+1,\eta-1}(k) - \frac{\varpi^i_{s,\eta}(k)}{\alpha^i_{s,\eta-1}(k)} \mathbf{f}^i_{s,\eta-1}(k), \quad s = 1 \cdots M-1. \end{aligned} \tag{3.39}$$

The proof of Equation 3.39 is included in Appendix A.2.

In terms of Equations 3.22 and 3.23, Equations 3.37 and 3.39 can be further reduced into

$$\varepsilon^i_{s,\eta}(k) = \varepsilon^i_{s,\eta-1}(k) - \frac{\psi^i_{s,\eta}(k)\gamma^i_{s+1,\eta-1}(k)}{\beta^i_{s+1,\eta-1}(k)}, \quad s = 1 \cdots M-1. \tag{3.40}$$

and

$$\gamma^i_{s,\eta}(k) = \gamma^i_{s+1,\eta-1}(k) - \frac{\varpi^i_{s,\eta}(k)\varepsilon^i_{s,\eta-1}(k)}{\alpha^i_{s,\eta-1}(k)}, \quad s = 1 \cdots M-1. \tag{3.41}$$

respectively.

The combination of Equations 3.26, 3.27, 3.31, 3.37 and 3.39 lead to

$$\begin{aligned}
\alpha^i_{s,\eta}(k) &= < \mathbf{f}^i_{s,\eta-1}(k-\mu) - \frac{\psi^i_{s,\eta}(k-\mu)}{\beta^i_{s+1,\eta-1}(k-\mu)}\mathbf{b}^i_{s+1,\eta-1}(k-\mu), \\
&\quad \mathbf{f}^i_{s,\eta-1}(k) - \frac{\psi^i_{s,\eta}(k)}{\beta^i_{s+1,\eta-1}(k)}\mathbf{b}^i_{s+1,\eta-1}(k) > \\
&= \alpha^i_{s,\eta-1}(k) - \frac{\varpi^i_{s,\eta}(k)\psi^i_{s,\eta}(k)}{\beta^i_{s+1,\eta-1}(k)}, \quad s = 1 \cdots M-1.
\end{aligned}$$

and

$$\begin{aligned}
\beta^i_{s,\eta}(k) &= < \mathbf{b}^i_{s+1,\eta-1}(k-\mu) - \frac{\varpi^i_{s,\eta}(k-\mu)}{\alpha^i_{s,\eta-1}(k-\mu)}\mathbf{f}^i_{s,\eta-1}(k-\mu), \\
&\quad \mathbf{b}^i_{s+1,\eta-1}(k) - \frac{\varpi^i_{s,\eta}(k)}{\alpha^i_{s,\eta-1}(k)}\mathbf{f}^i_{s,\eta-1}(k) > \\
&= \beta^i_{s+1,\eta-1}(k) - \frac{\varpi^i_{s,\eta}(k)\psi^i_{s,\eta}(k)}{\alpha^i_{s,\eta-1}(k)}, \quad s = 1 \cdots M-1.
\end{aligned}$$

However, the above order-recursion equations do not apply to the specific case $s = M$. With $s = M$, it turns out from Equations 3.16, 3.18 and 3.21 that

$$\mathbf{h}^i_{M,\eta}(k) = \mathbf{h}^i_{M,\eta-1}(k) \oplus \mathrm{Span}\{\mathbf{b}^i_{1,\eta-1}(k-1)\}$$

and

$$\mathbf{h}^i_{M-1,\eta}(k) = \mathrm{Span}\{\mathbf{f}^i_{M,\eta-1}(k)\} \oplus \mathbf{h}^i_{M,\eta-1}(k)$$

Using the above two equations in Equations 3.18 and 3.21, respectively, lead to

$$\mathbf{f}^i_{M,\eta}(k) = \mathbf{f}^i_{M,\eta-1}(k) - \frac{\psi^i_{M,\eta}(k)}{\beta^i_{1,\eta-1}(k-1)}\mathbf{b}^i_{1,\eta-1}(k-1)$$

and

$$\mathbf{b}^i_{M,\eta}(k) = \mathbf{b}^i_{1,\eta-1}(k-1) - \frac{\varpi^i_{M,\eta}(k)}{\alpha^i_{M,\eta-1}(k)}\mathbf{f}^i_{M,\eta-1}(k)$$

which can be further reduced to

$$\varepsilon^i_{M,\eta}(k) = \varepsilon^i_{M,\eta-1}(k) - \frac{\psi^i_{M,\eta}(k)\gamma^i_{1,\eta-1}(k-1)}{\beta^i_{1,\eta-1}(k-1)}$$

and

$$\gamma^i_{M,\eta}(k) = \gamma^i_{1,\eta-1}(k-1) - \frac{\varpi^i_{M,\eta}(k)\varepsilon^i_{M,\eta-1}(k)}{\alpha^i_{M,\eta-1}(k)}$$

Similarly, the order recursion equations for $\alpha^i_{M,\eta}(k)$ and $\beta^i_{M,\eta}(k)$ can be given by

$$\begin{aligned}\alpha^i_{M,\eta}(k) &= < \mathbf{f}^i_{M,\eta-1}(k-\mu) - \frac{\psi^i_{M,\eta}(k-\mu)}{\beta^i_{1,\eta-1}(k-1-\mu)}\mathbf{b}^i_{1,\eta-1}(k-\mu-1), \\ &\quad \mathbf{f}^i_{M,\eta-1}(k) - \frac{\psi^i_{M,\eta}(k)}{\beta^i_{1,\eta-1}(k-1)}\mathbf{b}^i_{1,\eta-1}(k-1) > \\ &= \alpha^i_{M,\eta-1}(k) - \frac{\varpi^i_{M,\eta}(k)\psi^i_{M,\eta}(k)}{\beta^i_{1,\eta-1}(k-1)}\end{aligned}$$

and

$$\begin{aligned}\beta^i_{M,\eta}(k) &= < \mathbf{b}^i_{1,\eta-1}(k-\mu-1) - \frac{\varpi^i_{M,\eta}(k-\mu)}{\alpha^i_{M,\eta-1}(k-\mu)}\mathbf{f}^i_{M,\eta-1}(k-\mu), \\ &\quad \mathbf{b}^i_{1,\eta-1}(k-1) - \frac{\varpi^i_{M,\eta}(k)}{\alpha^i_{M,\eta-1}(k)}\mathbf{f}^i_{M,\eta-1}(k) > \\ &= \beta^i_{1,\eta-1}(k-1) - \frac{\varpi^i_{M,\eta}(k)\psi^i_{M,\eta}(k)}{\alpha^i_{M,\eta-1}(k)}\end{aligned}$$

Finally, the order recursion equation for $\pi^i_{s,\eta+1}(k)$, which is defined in Equation 3.30, is derived as follows. As shown by Equation 3.36,

$$\mathbf{h}^i_{s-1,\eta+1}(k) = \mathbf{h}^i_{s-1,\eta}(k) \oplus \mathrm{Span}\{\mathbf{b}^i_{s,\eta}(k)\}$$

therefore, substituting $\mathbf{a}_k = \boldsymbol{\phi}$ and $\mathbf{h}_\eta(k) = \mathbf{h}^i_{s-1,\eta}(k) \oplus \mathrm{Span}\{\mathbf{b}^i_{s,\eta}(k)\}$ into Equations 3.24 and 3.25 results in

$$(\mathbf{a}_k)_{\mathbf{h}_\eta(k)} = \boldsymbol{\phi}_{\mathbf{h}^i_{s-1,\eta+1}(k)} = \boldsymbol{\phi}_{\mathbf{h}^i_{s-1,\eta}(k) \oplus \mathrm{Span}\{\mathbf{b}^i_{s,\eta}(k)\}}$$

Further, following the same step to derive Equation 3.37, the following equation is obtained

$$\boldsymbol{\phi}_{\mathbf{h}^i_{s-1,\eta+1}(k)} = \boldsymbol{\phi}_{\mathbf{h}^i_{s-1,\eta}(k)} - \mathbf{b}^i_{s,\eta}(k)\frac{\varsigma^i_{s,\eta}(k)}{\beta^i_{s,\eta}(k)} \tag{3.42}$$

where Equations 3.27 and 3.29, and the orthogonality between $\mathbf{\Lambda}\mathbf{b}^i_{s,\eta}(k)$ and $\mathbf{h}^i_{s-1,\eta}(k-\mu)$ are utilized. Therefore, substituting Equations 3.42 into 3.30 and performing a straightforward manipulation results in

$$\pi^i_{s,\eta+1}(k) = \pi^i_{s,\eta}(k) + \frac{\gamma^i_{s,\eta}(k)\varsigma^i_{s,\eta}(k)}{\beta^i_{s,\eta}(k)}, \quad s = 1 \cdots M.$$

where the order-recursion equation of $\varsigma^i_{s,\eta}(k)$ is

$$\varsigma^i_{s-1,\eta}(k) = \varsigma^i_{s,\eta-1}(k) - \frac{\psi^i_{s,\eta}(k)\epsilon^i_{s,\eta-1}(k)}{\alpha^i_{s,\eta-1}(k)} \tag{3.43}$$

with

$$\epsilon^i_{s,\eta}(k) = \begin{cases} \epsilon^i_{s,\eta-1}(k) - \frac{\varpi^i_{s,\eta}(k)\varsigma^i_{s+1,\eta-1}(k)}{\beta^i_{s+1,\eta-1}(k)} & s = 1 \cdots M-1. \\ \epsilon^i_{s,\eta-1}(k) - \frac{\varpi^i_{s,\eta}(k)\varsigma^i_{1,\eta-1}(k-1)}{\beta^i_{1,\eta-1}(k-1)} & s = M. \end{cases} \tag{3.44}$$

The derivation of Equations 3.43 and 3.44 is given in Appendix A.3, where another order recursion equation

$$\boldsymbol{\phi}_{\mathbf{h}^i_{s-1,\eta+1}(k)} = \boldsymbol{\phi}_{\mathbf{h}^i_{s,\eta}(k)} - \mathbf{f}^i_{s,\eta}(k)\frac{\epsilon^i_{s,\eta}(k)}{\alpha^i_{s,\eta}(k)} \tag{3.45}$$

is derived. Accordingly, by substituting Equations 3.45 into 3.30 and performing a straightforward manipulation the following equation is derived

$$\pi^i_{s,\eta+1}(k) = \begin{cases} \pi^i_{s+1,\eta}(k) + \frac{\epsilon^i_{s,\eta}(k)\epsilon^i_{s,\eta}(k)}{\alpha^i_{s,\eta}(k)} & s = 1 \cdots M-1. \\ \pi^i_{1,\eta}(k-1) + \frac{\epsilon^i_{s,\eta}(k)\epsilon^i_{s,\eta}(k)}{\alpha^i_{s,\eta}(k)} & s = M. \end{cases}$$

3.5.2 Time Recursion Equations

To compute the quantities $\{\varepsilon^i_{s,\eta}(k), \gamma^i_{s,\eta}(k), \alpha^i_{s,\eta}(k), \beta^i_{s,\eta}(k), \epsilon^i_{s,\eta}(k), \varsigma^i_{s,\eta}(k), \pi^i_{s,\eta}(k)\}$ recursively in order, $\varpi^i_{s,\eta}(k)$ and $\psi^i_{s,\eta}(k)$ must be known. The order-recursion equations for $\varpi^i_{s,\eta}(k)$ and $\psi^i_{s,\eta}(k)$ can not be developed. However they can be computed in a time-recursive manner, and the following equation is crucial for the derivation of the time-recursion equations. *For any vector* $\mathbf{a}_k$ *and subspace* $\mathbf{h}_\eta(k)$ *in the Hilbert space,*

$$(\mathbf{a}_k)_{(\mathbf{h}_\eta(k)\oplus\boldsymbol{\phi})} = \Big((\mathbf{a}_k)_{\boldsymbol{\phi}}\Big)_{\Big((\mathbf{h}_\eta(k))_{\boldsymbol{\phi}}\Big)} \tag{3.46}$$

The proof of Equation 3.46 is given in Appendix A.4.

With Equation 3.46, selecting $\mathbf{a}_k = \mathbf{z}^i_{kM-s}$ and $\mathbf{h}_\eta(k) = \mathbf{h}^i_{s,\eta}(k)$ and applying Equations 3.18, 3.24 and 3.25 give

$$\left(\mathbf{z}^i_{kM-s}\right)_{\boldsymbol{\phi}} = \begin{bmatrix} 0 \\ \mathbf{z}^i_{(k-1)M-s} \end{bmatrix}$$

$$\left(\mathbf{h}^i_{s,\eta}(k)\right)_{\boldsymbol{\phi}} = \begin{bmatrix} 0 \\ \mathbf{h}^i_{s,\eta}(k-1) \end{bmatrix}$$

and

$$\left(\left(\mathbf{z}^{\mathbf{i}}_{kM-s}\right)_{\boldsymbol{\phi}}\right)_{\left(\left(\mathbf{h}^{\mathbf{i}}_{s,\eta}(k)\right)_{\boldsymbol{\phi}}\right)} = \left(\begin{bmatrix} 0 \\ \mathbf{z}^{\mathbf{i}}_{(k-1)M-s} \end{bmatrix}\right)\left(\begin{bmatrix} 0 \\ \mathbf{h}^{\mathbf{i}}_{s,\eta}(k-1) \end{bmatrix}\right)$$
$$= \begin{bmatrix} 0 \\ \mathbf{f}^{\mathbf{i}}_{s,\eta}(k-1) \end{bmatrix}$$

On the other hand, noting that

$$\mathbf{h}^{\mathbf{i}}_{s,\eta}(k) \oplus \boldsymbol{\phi} = \mathbf{h}^{\mathbf{i}}_{s,\eta}(k) \oplus \boldsymbol{\phi}_{\mathbf{h}^{\mathbf{i}}_{s,\eta}(k)}$$

and re-applying Equations 3.18, 3.24 and 3.25, it can be observed that

$$\left(\mathbf{z}^{\mathbf{i}}_{kM-s}\right)_{\mathbf{h}^{\mathbf{i}}_{s,\eta}(k)\oplus\boldsymbol{\phi}_{\mathbf{h}^{\mathbf{i}}_{s,\eta}(k)}} = \mathbf{f}^{\mathbf{i}}_{s,\eta}(k) - \frac{\varepsilon^{\mathbf{i}}_{s,\eta}(k)}{1-\pi^{\mathbf{i}}_{s+1,\eta}(k)}\boldsymbol{\phi}_{\mathbf{h}^{\mathbf{i}}_{s,\eta}(k)}, \quad s = 1 \cdots M-1. \tag{3.47}$$

where the derivation is exactly similar to that of Equation 3.37 as derived in Appendix A.1. Therefore, according to Equation 3.46, it is concluded that

$$\mathbf{f}^{\mathbf{i}}_{s,\eta}(k) = \begin{bmatrix} 0 \\ \mathbf{f}^{\mathbf{i}}_{s,\eta}(k-1) \end{bmatrix} + \frac{\varepsilon^{\mathbf{i}}_{s,\eta}(k)}{1-\pi^{\mathbf{i}}_{s+1,\eta}(k)}\boldsymbol{\phi}_{\mathbf{h}^{\mathbf{i}}_{s,\eta}(k)}, \quad s = 1 \cdots M-1. \tag{3.48}$$

In the specific case $s = M$, using the similar steps shown above, it can be proved that

$$\mathbf{f}^{\mathbf{i}}_{M,\eta}(k) = \begin{bmatrix} 0 \\ \mathbf{f}^{\mathbf{i}}_{M,\eta}(k-1) \end{bmatrix} + \frac{\varepsilon^{\mathbf{i}}_{M,\eta}(k)}{1-\pi^{\mathbf{i}}_{1,\eta}(k-1)}\boldsymbol{\phi}_{\mathbf{h}^{\mathbf{i}}_{M,\eta}(k)} \tag{3.49}$$

The choice of $\mathbf{a}_k = \mathbf{z}^{\mathbf{i}}_{kM-s-\eta}$ and $\mathbf{h}_\eta(k) = \mathbf{h}^{\mathbf{i}}_{s-1,\eta}(k)$ gives

$$\left(\mathbf{z}^{\mathbf{i}}_{kM-s-\eta}\right)_{\boldsymbol{\phi}} = \begin{bmatrix} 0 \\ \mathbf{z}^{\mathbf{i}}_{(k-1)M-s-\eta} \end{bmatrix}$$

$$\left(\mathbf{h}^{\mathbf{i}}_{s-1,\eta}(k)\right)_{\boldsymbol{\phi}} = \begin{bmatrix} 0 \\ \mathbf{h}^{\mathbf{i}}_{s-1,\eta}(k-1) \end{bmatrix}$$

and

$$\left(\left(\mathbf{z}^{\mathbf{i}}_{kM-s-\eta}\right)_{\boldsymbol{\phi}}\right)_{\left(\left(\mathbf{h}^{\mathbf{i}}_{s-1,\eta}(k)\right)_{\boldsymbol{\phi}}\right)} = \left(\begin{bmatrix} 0 \\ \mathbf{z}^{\mathbf{i}}_{(k-1)M-s-\eta} \end{bmatrix}\right)\left(\begin{bmatrix} 0 \\ \mathbf{h}^{\mathbf{i}}_{s-1,\eta}(k-1) \end{bmatrix}\right)$$
$$= \begin{bmatrix} 0 \\ \mathbf{b}^{\mathbf{i}}_{s,\eta}(k-1) \end{bmatrix}$$

where Equation 3.21 has been taken into account. Since

$$\mathbf{h}^{\mathbf{i}}_{s-1,\eta}(k) \oplus \boldsymbol{\phi} = \mathbf{h}^{\mathbf{i}}_{s-1,\eta}(k) \oplus \boldsymbol{\phi}_{\mathbf{h}^{\mathbf{i}}_{s-1,\eta}(k)}$$

using Equations 3.24 and 3.25 again, it can be easily shown that

$$\left(\mathbf{z}^i_{kM-s-\eta}\right)_{\mathbf{h}^i_{s-1,\eta}(k)\oplus\boldsymbol{\phi}_{\mathbf{h}^i_{s-1,\eta}(k)}} = \mathbf{b}^i_{s,\eta}(k) - \frac{\gamma^i_{s,\eta}(k)}{1-\pi^i_{s,\eta}(k)}\boldsymbol{\phi}_{\mathbf{h}^i_{s-1,\eta}(k)}$$

where the detailed derivation is similar to that of Equation 3.37. Further, Equation 3.46 leads to

$$\mathbf{b}^i_{s,\eta}(k) = \begin{bmatrix} 0 \\ \mathbf{b}^i_{s,\eta}(k-1) \end{bmatrix} + \frac{\gamma^i_{s,\eta}(k)}{1-\pi^i_{s,\eta}(k)}\boldsymbol{\phi}_{\mathbf{h}^i_{s-1,\eta}(k)}, \quad s = 1\cdots M. \tag{3.50}$$

With Equations 3.48, 3.49 and 3.50, deriving the time-recursion equations for $\alpha^i_{s,\eta}(k)$, $\beta^i_{s,\eta}(k)$, $\varpi^i_{s,\eta}(k)$, and $\psi^i_{s,\eta}(k)$ is straightforward. For instance, using Equation 3.48 in Equation 3.26 results in

$$\begin{aligned}
[0|\ \left(\mathbf{f}^i_{s,\eta}(k-1-\mu)\right)^T]\boldsymbol{\Lambda}\begin{bmatrix} 0 \\ \mathbf{f}^i_{s,\eta}(k-1) \end{bmatrix} &= \lambda\alpha^i_{s,\eta}(k-1) \\
&= < \mathbf{f}^i_{s,\eta}(k-\mu) - \frac{\varepsilon^i_{s,\eta}(k-\mu)}{1-\pi^i_{s+1,\eta}(k-\mu)}\boldsymbol{\phi}_{\mathbf{h}^i_{s,\eta}(k-\mu)}, \\
&\quad \mathbf{f}^i_{s,\eta}(k) - \frac{\varepsilon^i_{s,\eta}(k)}{1-\pi^i_{s+1,\eta}(k)}\boldsymbol{\phi}_{\mathbf{h}^i_{s,\eta}(k)} > \\
&= \alpha^i_{s,\eta}(k) - \frac{\varepsilon^i_{s,\eta}(k)\epsilon^i_{s,\eta}(k)}{1-\pi^i_{s+1,\eta}(k)}, \quad s = 1\cdots M-1.
\end{aligned}$$

where Equations 3.28 and 3.30 and $< \boldsymbol{\phi}_{\mathbf{h}^i_{s,\eta}(k-\mu)}, \mathbf{f}^i_{s,\eta}(k) > = \varepsilon^i_{s,\eta}(k)$ are utilized. Therefore,

$$\alpha^i_{s,\eta}(k) = \lambda\alpha^i_{s,\eta}(k-1) + \frac{\varepsilon^i_{s,\eta}(k)\epsilon^i_{s,\eta}(k)}{1-\pi^i_{s+1,\eta}(k)}, \quad s = 1\cdots M-1.$$

Similarly, following equation is also obtained

$$\alpha^i_{M,\eta}(k) = \lambda\alpha^i_{M,\eta}(k-1) + \frac{\varepsilon^i_{M,\eta}(k)\epsilon^i_{M,\eta}(k)}{1-\pi^i_{1,\eta}(k-1)}$$

Moreover, using Equation 3.50 in Equation 3.27 gives

$$\beta^i_{s,\eta}(k) = \lambda\beta^i_{s,\eta}(k-1) + \frac{\gamma^i_{s,\eta}(k)\varsigma^i_{s,\eta}(k)}{1-\pi^i_{s,\eta}(k)}, \quad s = 1\cdots M. \tag{3.51}$$

Finally, substituting Equations 3.48, 3.49 and 3.50 into 3.31 and 3.32 respectively, the following time recursion equations are obtained

$$\varpi^i_{s,\eta}(k) = \begin{cases} \lambda\varpi^i_{s,\eta}(k-1) + \frac{\epsilon^i_{s,\eta-1}(k)\gamma^i_{s+1,\eta-1}(k)}{1-\pi^i_{s+1,\eta-1}(k)} & s = 1\cdots M-1. \\ \lambda\varpi^i_{s,\eta}(k-1) + \frac{\epsilon^i_{s,\eta-1}(k)\gamma^i_{1,\eta-1}(k-1)}{1-\pi^i_{1,\eta-1}(k-1)} & s = M. \end{cases}$$

and

$$\psi^i_{s,\eta}(k) = \begin{cases} \lambda\psi^i_{s,\eta}(k-1) + \frac{\varsigma^i_{s+1,\eta-1}(k)\varepsilon^i_{s,\eta-1}(k)}{1-\pi^i_{s+1,\eta-1}(k)} & s = 1\cdots M-1. \\ \lambda\psi^i_{s,\eta}(k-1) + \frac{\varsigma^i_{1,\eta-1}(k-1)\varepsilon^i_{s,\eta-1}(k)}{1-\pi^i_{1,\eta-1}(k-1)} & s = M. \end{cases}$$

where the relationship

$$\begin{aligned} < \mathbf{b}^i_{s+1,\eta-1}(k), \phi_{\mathbf{h}^i_{s,\eta-1}(k-\mu)} > &= \gamma^i_{s+1,\eta-1}(k) \\ < \mathbf{f}^i_{s,\eta-1}(k), \phi_{\mathbf{h}^i_{s,\eta-1}(k)} > &= \varepsilon^i_{s,\eta-1}(k) \end{aligned}$$

and the definitions of $\{\epsilon^i_{s,\eta}(k)$, $\varsigma^i_{s,\eta}(k)$, and $\pi^i_{s,\eta+1}(k)\}$ have been utilized.

3.5.3 Parameter Recursion Equations

With the above order and time recursion equations, algorithm for recursive identification of the parameter vector $\boldsymbol{\theta}^i_n(k)$ can be readily derived as follows.

Introducing two auxiliary parameter vectors: $\boldsymbol{\vartheta}^i_{s,\eta}(k) \in \Re^\eta$ and $\boldsymbol{\varrho}^i_{s,\eta}(k) \in \Re^\eta$ for $i = 1\cdots m$ and $\eta = 1\cdots M_n - 1$, it follows that

$$\varepsilon^i_{s,\eta}(k) = z^i(kM-s) - \left(\boldsymbol{\vartheta}^i_{s,\eta}(k)\right)^T \begin{bmatrix} z^i(kM-s-1) \\ \vdots \\ z^i(kM-s-\eta) \end{bmatrix} \tag{3.52}$$

and

$$\gamma^i_{s,\eta}(k) = z^i(kM-s-\eta) - \left(\boldsymbol{\varrho}^i_{s,\eta}(k)\right)^T \begin{bmatrix} z^i(kM-s) \\ \vdots \\ z^i(kM-s-\eta+1) \end{bmatrix} \tag{3.53}$$

where

$$\boldsymbol{\vartheta}^i_{1,M_n-1}(k) \equiv \boldsymbol{\theta}^i_n(k)$$

Using Equations 3.52 and 3.53 in Equations 3.40 and 3.41 result in

$$\boldsymbol{\vartheta}^i_{s,\eta}(k) = \left[\left(\boldsymbol{\vartheta}^{i,}_{s,\eta-1}(k)\right)^T \mid 0\right]^T - \frac{\psi^i_{s,\eta}(k)}{\beta^i_{s+1,\eta-1}(k)}\left[\left(\boldsymbol{\varrho}^i_{s+1,\eta-1}(k)\right)^T \mid -1\right]^T, \quad s = 1\cdots M-1.$$

and

$$\boldsymbol{\varrho}^i_{s,\eta}(k) = \left[0 \mid \left(\boldsymbol{\varrho}^i_{s+1,\eta-1}(k)\right)^T\right]^T - \frac{\varpi^i_{s,\eta}(k)}{\alpha^i_{s,\eta-1}(k)}\left[-1 \mid \left(\boldsymbol{\vartheta}^i_{s,\eta-1}(k)\right)^T\right]^T, \quad s = 1\cdots M-1.$$

Similarly, with $s = M$

$$\boldsymbol{\vartheta}^i_{M,\eta}(k) = \left[\left(\boldsymbol{\vartheta}^i_{M,\eta-1}(k)\right)^T \mid 0\right]^T - \frac{\psi^i_{M,\eta}(k)}{\beta^i_{1,\eta-1}(k-1)}\left[\left(\boldsymbol{\varrho}^i_{1,\eta-1}(k-1)\right)^T \mid -1\right]^T$$

and

$$\varrho^{i}_{M,\eta}(k) = \left[0|\ \left(\varrho^{i}_{1,\eta-1}(k-1)\right)^{T}\right]^{T} - \frac{\varpi^{i}_{M,\eta}(k)}{\alpha^{i}_{M,\eta-1}(k)}\left[-1|\ \left(\vartheta^{i}_{M,\eta-1}(k)\right)^{T}\right]^{T}$$

Finally, the complete lattice filter algorithms for the recursive identification of $\theta^{i}_{n}(k)$ are listed in Appendix E for easy reference.

3.6 Lattice filter based Process Monitoring

As discussed previously, the proposed IV lattice filter can recursively compute

$$\mathbf{e}_n(k) \equiv \begin{bmatrix} \varepsilon^{1}_{1,M_n-1}(k) \\ \vdots \\ \varepsilon^{m}_{1,M_n-1}(k) \end{bmatrix} \equiv \boldsymbol{\varepsilon}_{1,M_n-1}(k)$$

This residual error signal is now used to construct a statistic for process monitoring.

3.6.1 Initialization

The process data is assumed to be available only for $k \geq 0$. Therefore, there is a need to initialize the lattice filtering algorithms at a certain time instant. This can be done by assigning initial values to the quantities $\alpha^{i}_{s,0}(k)$, $\varpi^{i}_{s,\eta}(k)$, and $\psi^{i}_{s,\eta}(k)$.

The initialization procedure is listed as follows:

1. Assign initial values to the quantities $\alpha^{i}_{s,0}(Int\left(\frac{s}{M}\right)+\mu-1)$, $\varpi^{i}_{s,\eta}(Int\left(\frac{s+\eta-1}{M}\right)+2\mu-1)$, and $\psi^{i}_{s,\eta}(Int\left(\frac{s+\eta-1}{M}\right)+2\mu-1)$ for $s \in [1 \cdots M]$ and $\eta \in [1 \cdots M_n - 1]$.
2. For any $2\mu < k < Int\left(\frac{M+M_n-2}{M}\right)+2\mu$ and $s = 1 \cdots M$, execute the complete lattice filtering algorithm for $\eta \in [1 \cdots (k-2\mu)M - s + 1]$.
3. When $k \geq Int\left(\frac{M+M_n-2}{M}\right)+2\mu$, execute the complete algorithm for $\eta \in [1 \cdots M_n - 1]$.

Note that the operator $Int(\cdot)$ means that the argument has to be converted into an integer.

3.6.2 Monitoring Index

As shown by the last line of Equation 3.13, under the normal case $\mathbf{e}_n(k)$ is a moving average process with simultaneous excitation by noise vectors $\mathbf{o}(k)$, $\mathbf{v}(k)$ and $\mathbf{p}(k)$. If any of these noise vectors is a multivariate zero-mean Gaussian process, it follows that

$$\mathbf{e}_n(k) \sim \aleph(\mathbf{0}, \mathbf{R}^{e}_{n}(k))$$

and consequently

$$\mathbf{e}^{T}_{n}(k)\left(\mathbf{R}^{e}_{n}(k)\right)^{-1}\mathbf{e}_n(k) \sim \chi^2(m)$$

However, the true value of $\mathbf{R}_n^e(k)$ is usually unknown. To work around this issue, Hotelling T^2 statistic is used as the index for process monitoring, which is based on the estimated covariance matrix. Moreover, $\mathbf{e}_n(k)$ being an n^{th} order MA process, it is auto-correlated with $\mathbf{e}_n(k+\tau)$ for any $0 \leq \tau \leq n$. Therefore, at the k^{th} time instant, the following residual vectors sequence

$$\{\mathbf{e}_n(0)\ \mathbf{e}_n(1) \cdots \mathbf{e}_n(k-1)\ \mathbf{e}_n(k)\}$$

can not be directly used to compute the T^2 statistic (Johnson and Wichern, 1998). To work with a series of uncorrelated signals from the aforementioned sequence, a series of lagged signals are considered, i.e.,

$$\{\mathbf{e}_n(0)\ \mathbf{e}_n(\tau) \cdots \mathbf{e}_n(Int(\frac{k-\tau}{\tau})\tau)\ \mathbf{e}_n(Int(\frac{k}{\tau})\tau)\}$$

for the computation of the T^2 statistic. Clearly the disadvantage of this approach is the time delay in detecting a fault during monitoring. For instance, if the process has an abnormal variation at time instant k, then such a variation will not be detected until the time instant $k+\tau$. Generally, since τ is selected to be $n+1$, if the process order n is not high, a time delay of this magnitude can be tolerated.

At time instant k_0, assume that a sequence of uncorrelated residual vectors:

$$\{\mathbf{e}_n(0)\ \mathbf{e}_n(\tau) \cdots \mathbf{e}_n(Int(\frac{k_0}{\tau})\tau - \tau)\ \mathbf{e}_n(Int(\frac{k_0}{\tau})\tau)\}$$

has been computed when a process is normal. Now the goal is to use this residual sequence to determine a nominal operation or control region as a template for comparing with future residuals:

$$\{\mathbf{e}_n(Int(\frac{k_0}{\tau})\tau + \tau)\ \mathbf{e}_n(Int(\frac{k_0}{\tau})\tau + 2\tau) \cdots \mathbf{e}_n(Int(\frac{k}{\tau})\tau)\}$$

Denoting $\kappa_1 \equiv Int(\frac{k_0}{\tau})\tau$ and selecting $Int(\frac{k_0}{\tau})$ uncorrelated residual vectors to compute

$$\bar{\mathbf{e}}_n(\kappa_1) = \frac{1}{Int(\frac{k_0}{\tau})} \sum_{i=0}^{Int(\frac{k_0}{\tau})-1} \mathbf{e}_n(\kappa_1 + (i - Int(\frac{k_0}{\tau}) + 1)\tau) \tag{3.54}$$

with $\tau = n+1$. According to Johnson and Wichern (1998),

$$T_n^2(k) \equiv \frac{Int(\frac{k_0}{\tau})}{Int(\frac{k_0}{\tau})+1} (\mathbf{e}_n(k) - \bar{\mathbf{e}}_n(\kappa_1))^T \left(\hat{\mathbf{R}}_n^e(\kappa_1)\right)^{-1} (\mathbf{e}_n(k) - \bar{\mathbf{e}}_n(\kappa_1)) \tag{3.55}$$

is distributed as

$$\frac{(Int(\frac{k_0}{\tau})-1)m}{Int(\frac{k_0}{\tau})-m} F_{m,Int(\frac{k_0}{\tau})-m}$$

where (1) $F_{m,Int(\frac{k_0}{\tau})-m}$ stands for an F-distributed random variable with degrees of freedom

m and $Int(\frac{k_0}{\tau}) - m$; and (2)

$$\hat{\mathbf{R}}_n^e(\kappa_1) = \frac{1}{Int(\frac{k_0}{\tau}) - 1} \sum_{i=0}^{Int(\frac{k_0}{\tau})-1} \left(\mathbf{e}_n(\kappa_1 + (i - Int(\frac{k_0}{\tau}) + 1)\tau) - \bar{\mathbf{e}}_n(\kappa_1)\right) \bullet \left(\mathbf{e}_n(\kappa_1 + (i - Int(\frac{k_0}{\tau}) + 1)\tau) - \bar{\mathbf{e}}_n(\kappa_1)\right)^T \tag{3.56}$$

is the estimate of covariance matrix of $\mathbf{e}_n(t)$ based on a sequence of $Int(\frac{k_0}{\tau})$ uncorrelated residual vectors.

Therefore, given a level of significance ϕ, the confidence limit of $T_n^2(k)$ for process monitoring will be

$$\frac{(Int(\frac{k_0}{\tau}) - 1)m}{Int(\frac{k_0}{\tau}) - m} F_{m, Int(\frac{k_0}{\tau})-m}(\phi)$$

which is a function of $Int(\frac{k_0}{\tau})$ and m, the dimension of the residual vector $\mathbf{e}_n(\bullet)$.

As each newly computed residual vector becomes available, $T_n^2(k)$ can be computed and checked whether it stays within the control region. In this way the process under consideration can be effectively monitored.

3.6.3 Treatment of Some Practical Issues

When the proposed approach to monitoring a practical process is applied, some practical issues need to be considered.

Pre-processing of Process Data

The process variables sampled at the t^{th} instant are denoted by $\mathbf{z}(t)$ which includes both inputs and outputs. First of all, outliers in the data should be removed. Many data sets may contain unusual observations that do not seem to belong to the pattern of variability produced by the other observations. These unusual observations are referred to as *outliers*. For data with a single characteristic, outliers are those that are either very large or very small relative to the others. The situation can be more complicated with multivariate data. Outliers are best detected visually whenever this is possible. When the number of observations k is large, scatter plots are not feasible. When the number of variables $m + l$ (number of inputs and outputs) is large, the large number of scatter plots $(m+l)(m+l-1)/2$ may prevent viewing them all. But still it is suggested to have visual inspection of the data whenever possible. If the process variables are high-dimensional, outliers can not be detected from the univariate or bivariate scatter plots easily. However, in this case, a large value of $(\mathbf{z}(t) - \bar{\mathbf{z}})^T (\mathbf{S})^{-1} (\mathbf{z}(t) - \bar{\mathbf{z}})$ for $t \in [0, k]$ will suggest an unusual observation (Johnson and Wichern, 1998), even though it can not be seen visually, where $\bar{\mathbf{z}}$ and $\mathbf{S}$ are the mean and covariance matrix of $\mathbf{z}$ estimated over $t \in [0, k]$.

Secondly, some process data may be missing. Since correlations exist among multivariate data of process variables, these correlations can be used to reconstruct the missing data. For example, the approach proposed by Dunia et al. (1996) and Burnham et al. (1999) can be used for data reconstruction.

Finally, there is a need to ensure that each output variable is linearly independent of the others. For real data, if some elements of $\mathbf{y}(k)$ are correlated, data pre-processing should be carried out. For example, PCA can be performed on the sampled $\mathbf{y}(k)$ to remove its correlated elements. Consequently, instead of $\mathbf{y}(k)$, the scores $\hat{\mathbf{p}}_y^T\mathbf{y}(k)$ will be the inputs to the lattice filter, where $\hat{\mathbf{p}}_y$ are the principal eigenvectors of the covariance matrix of $\mathbf{y}(k)$. Similarly the sampled input, $\mathbf{u}(k)$, can be pre-processed if it has correlated elements so that not $\mathbf{u}(k)$ but its scores $\hat{\mathbf{p}}_u^T\mathbf{u}(k)$ will be the to the lattice filter, where $\hat{\mathbf{p}}_u$ are the principal eigenvectors of the covariance matrix of $\mathbf{u}(k)$. Alternately, data pre-filtering can also be used to remove the correlation.

On-line Recursive Order Determination for Lattice Filters

When applying the proposed lattice filter-based approach to the monitoring of a process, since the process order n is usually unknown *a priori* and can vary, the process order has to be estimated in real time and on-line.

The correct choice of a process order is very important. If the chosen order is too low, the process will be underparameterized, and as a consequence, the unmodeled dynamics of the process may destabilize. This means that in practice there maybe a tendency to overparameterize the process. However, with overparameterization, there is a danger of ill-conditioning in the lattice filter algorithm and the convergence of the algorithm can not be guaranteed (Xia and Moore, 1989).

Chen and Guo (1987) have proposed an approach based on Bayesian Information Criterion (BIC) for the consistent estimation of the order of a class of stochastic systems given a batch of data. BIC approach is adopted for the on-line recursive determination of the order of the lattice filter.

From $\mathbf{e}_n(k) \equiv \boldsymbol{\varepsilon}_{1,M_n-1}(k)$, it turns out that the process order n corresponds to the order index $\eta = M_n - 1$ of the lattice filter. Therefore, the determination of n is in a one-to-one correspondence with the determination of $\eta = M_n - 1$.

With an available sequence of residual vectors $\{\boldsymbol{\varepsilon}_{1,M_n-1}(0)\cdots\boldsymbol{\varepsilon}_{1,M_n-1}(k)\}$, define a function

$$\Omega_k(\eta) = klog\left(\sum_{t=0}^{k}\boldsymbol{\varepsilon}_{1,\eta}^T(t)\boldsymbol{\varepsilon}_{1,\eta}(t)\right) + 2\left(\frac{\eta+1}{M} - 1\right)\log(k)\log\left(\log(k)\right) \tag{3.57}$$

Then the lattice filter order η_0 will be determined by minimizing $\Omega_k(\eta)$ with respect to $\eta \in [M-1,\cdots,M_L-1]$, e.g.,

$$\eta_0 = \arg\min\left(\Omega_k(\eta)\right), \quad \eta \in [M-1\cdots M_L-1] \tag{3.58}$$

where L can be a large positive integer and $M_L = M(L+1)$. Moreover, since the process order n can vary, there is a need to develop a recursive scheme for order determination as new data become available.

Intuitively, for a time-varying process, its order change is less frequent than its parameter change. Therefore, a batch-wise approach is proposed for recursive order determination. Define an intermediate variable

$$\Xi_{\boldsymbol{\varepsilon}}(k) \equiv \sum_{t=0}^{k} \boldsymbol{\varepsilon}_{1,\eta}^{T}(t)\boldsymbol{\varepsilon}_{1,\eta}(t) \tag{3.59}$$

Rewriting Equation 3.57 into

$$\Omega_k(\eta) = k \log \Xi_{\boldsymbol{\varepsilon}}(k) + 2\left(\frac{\eta+1}{M} - 1\right)\log(k)\log\left(\log(k)\right) \tag{3.60}$$

If at time instant $k+n_{k+1}$, a new sequence of residual vectors $\{\boldsymbol{\varepsilon}_{1,M_n-1}(k+1)\cdots\boldsymbol{\varepsilon}_{1,M_n-1}(k+n_{k+1})\}$ becomes available ($n_{k+1} \geq 1$), then in terms of Equation 3.59

$$\Xi_{\boldsymbol{\varepsilon}}(k+n_{k+1}) \equiv \Xi_{\boldsymbol{\varepsilon}}(k) + \sum_{t=k+1}^{k+n_{k+1}} \boldsymbol{\varepsilon}_{1,\eta}^{T}(t)\boldsymbol{\varepsilon}_{1,\eta}(t) \tag{3.61}$$

Therefore,

$$\begin{aligned}\Omega_{k+n_{k+1}}(\eta) &= (k+n_{k+1})log\left(\Xi_{\boldsymbol{\varepsilon}}(k) + \sum_{t=k+1}^{k+n_{k+1}} \boldsymbol{\varepsilon}_{1,\eta}^{T}(t)\boldsymbol{\varepsilon}_{1,\eta}(t)\right) \\ &+ 2\left(\frac{\eta+1}{M} - 1\right)\log(k+n_{k+1})\log\left(\log(k+n_{k+1})\right)\end{aligned} \tag{3.62}$$

Using Equation 3.61, $\Omega_{k+n_{k+1}}(\eta)$ is recursively computed. Eventually, as shown in Equation 3.58 if the η_0 corresponding to the minimum of $\Omega_{k+n_{k+1}}(\eta)$ is found, then the new process order will be updated to:

$$n_0 = \frac{\eta_0+1}{M} - 1$$

Update of the Covariance Matrix

Since the process under surveillance is time-varying, the covariance matrix $\hat{\mathbf{R}}_n^e(Int(\frac{k_0}{\tau})\tau)$ is recursively updated based on the newly sampled process data. This will provide the proposed monitoring scheme an adaptive capability. As the time instant k progresses, some old data should be discounted and the most recent data should be *weighted* heavily in the recursive updates. When the covariance matrix is updated using a sequence of w_l

uncorrelated residual vectors at one time, at the time instant $\kappa_2 = \kappa_1 + w_l\tau$,

$$\begin{aligned}\hat{\mathbf{R}}_n^e(\kappa_2) &= \frac{\frac{\kappa_1}{\tau}-1}{\frac{\kappa_2}{\tau}-1}\hat{\mathbf{R}}_n^e(\kappa_1) + \frac{\frac{\kappa_1}{\tau}}{\frac{\kappa_2}{\tau}-1}\left(\bar{\mathbf{e}}_n(\kappa_1) - \bar{\mathbf{e}}_n^T(\kappa_2)\right)\left(\bar{\mathbf{e}}_n(\kappa_1) - \bar{\mathbf{e}}_n^T(\kappa_2)\right)^T \\ &+ \frac{1}{\frac{\kappa_2}{\tau}-1}\sum_{i=\frac{\kappa_2}{\tau}-w_l}^{\frac{\kappa_2}{\tau}-1}\left(\mathbf{e}_n(\kappa_1 + (i - \frac{\kappa_2}{\tau} + 1 + w_l)\tau) - \bar{\mathbf{e}}_n(\kappa_1)\right)\bullet \\ &\qquad \left(\mathbf{e}_n(\kappa_1 + (i - \frac{\kappa_2}{\tau} + 1 + w_l)\tau) - \bar{\mathbf{e}}_n(\kappa_1)\right)^T \end{aligned} \quad (3.63)$$

and the mean update equation is

$$\bar{\mathbf{e}}_n(\kappa_2) = \frac{\kappa_1}{\kappa_2}\bar{\mathbf{e}}_n(\kappa_1) + \frac{\tau}{\kappa_2}\sum_{i=\frac{\kappa_2}{\tau}-w_l}^{\frac{\kappa_2}{\tau}-1}\mathbf{e}_n(\kappa_1 + (i - \frac{\kappa_2}{\tau} + 1 + w_l)\tau) \quad (3.64)$$

The derivation of Equations 3.63 and 3.64 is simple and straightforward.

3.6.4 Adaptive Process Monitoring

Once the data pre-processing is completed, the sequence of steps required for real-time and on-line adaptive process monitoring is as follows.

1. Choose an initial data block with d_0 samples and apply the proposed lattice filtering algorithm given in Appendix A.5 to the first segment of the block to generate residual vectors. Determine the order of the lattice filter from the generated residual vectors using the index given in Equation 3.60.

2. With the determined n, let τ be $n+1$. Apply the lattice filter to the remaining data in the initial data block. Select $Int(\frac{d_0}{\tau})$ uncorrelated residual vectors to estimate the covariance matrix $\mathbf{R}_n^e(Int(\frac{d_0}{\tau})\tau)$ according to Equation 3.56.

3. As new data samples become available, calculate the residual vectors based on the lattice filters. Select w_l, and Calculate $T_n^2(k)$ and compare it with its pre-determined confidence limit. If $T_n^2(k)$ is within its confidence limit, recursively update the covariance matrix with w_l uncorrelated residual vectors and the order with n_{k+1} residual vectors, according to Equations 3.62 and 3.63, respectively. Otherwise, stop updating, announce alarm, and take a further action for diagnosis if necessary.

4. The aforementioned monitoring procedure is repeated for each new data sample.

3.7 Case Studies

In this section, two case studies are carried out to illustrate the application of the proposed lattice filter algorithm to dynamic process monitoring. The objective of the studies is to show the lattice filter's ability to adapt to slow, normal process changes and detect abnormal operations. The first case study is based on a simulated process, and the second one is based on a real pilot scale plant.

3.7.1 Case Study One- Simulation Example

A second order dynamic process with four inputs and four outputs, e.g.,

$$\begin{aligned}
\mathbf{x}(k+1) &= \begin{bmatrix} 0.67 & 0.67 \\ -0.67 & 0.67 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} -0.4326 & 0.1253 & -1.1456 & 1.1892 \\ -1.6656 & 0.2877 & 1.1909 & -0.0376 \end{bmatrix} \tilde{\mathbf{u}}(k) + \mathbf{p}(k) \\
\tilde{\mathbf{y}}(k) &= \begin{bmatrix} 0.3273 & -0.5883 \\ 0.1746 & 2.1832 \\ -0.1867 & -0.1364 \\ 0.7258 & 0.1139 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 1.0668 & 0.2944 & -0.6918 & -1.4410 \\ 0.0593 & -1.3362 & 0.8580 & 0.5711 \\ -0.0956 & 0.7143 & 1.2540 & -0.3999 \\ -0.8323 & 1.6236 & -1.5937 & 0.6900 \end{bmatrix} \tilde{\mathbf{u}}(k)
\end{aligned} \tag{3.65}$$

is used to generate simulated data. Further, according to Equation 3.8, Equation 3.65 is converted into the following input-output equation:

$$\begin{aligned}
\tilde{\mathbf{y}}(k) - 1.34\tilde{\mathbf{y}}(k-1) + 0.8978\tilde{\mathbf{y}}(k-2) &= \begin{bmatrix} 1.0668 & 0.2944 & -0.6918 & -1.4410 \\ 0.0593 & -1.3362 & 0.8580 & 0.5711 \\ -0.0956 & 0.7143 & 1.2540 & -0.3999 \\ -0.8323 & 1.6236 & -1.5937 & 0.6900 \end{bmatrix} \tilde{\mathbf{u}}(k) \\
&+ \begin{bmatrix} -0.5912 & -0.5227 & -0.1485 & 2.3423 \\ -3.7913 & 2.4405 & 1.2502 & -0.6397 \\ 0.4361 & -1.0198 & -1.6289 & 0.3190 \\ 0.6116 & -2.0519 & 1.4397 & -0.0658 \end{bmatrix} \tilde{\mathbf{u}}(k-1) \\
&+ \begin{bmatrix} -0.1396 & 0.4627 & -0.0909 & -1.1088 \\ 2.9781 & -1.7848 & 0.9774 & -1.3153 \\ -0.1233 & 0.6587 & 0.8377 & -0.1003 \\ -1.1867 & 1.5051 & -0.2981 & -0.0650 \end{bmatrix} \tilde{\mathbf{u}}(k-2) \\
&+ \begin{bmatrix} 0.3273 & -0.5883 & 0.1749 & 0.6135 \\ 0.1746 & 2.1832 & -1.5797 & -1.3458 \\ -0.1867 & -0.1364 & 0.2165 & -0.0337 \\ 0.7258 & 0.1139 & -0.5626 & 0.4100 \end{bmatrix} \begin{bmatrix} \mathbf{p}(k-1) \\ \mathbf{p}(k-2) \end{bmatrix}
\end{aligned} \tag{3.66}$$

A combination of sinewaves with different frequencies as the noise-free input sequence $\{\tilde{\mathbf{u}}(k)\}$ is selected. Also it is assumed that the observed inputs $\mathbf{u}(k)$ and outputs $\mathbf{y}(k)$ are corrupted by noise vectors $\mathbf{o}(k)$ and $\mathbf{v}(k)$ respectively, e.g.,

$$\begin{aligned}
\mathbf{u}(k) &= \tilde{\mathbf{u}}(k) + \mathbf{v}(k) \\
\mathbf{y}(k) &= \tilde{\mathbf{y}}(k) + \mathbf{o}(k)
\end{aligned} \tag{3.67}$$

where $\mathbf{o}(k)$, $\mathbf{v}(k)$ and $\mathbf{p}(k)$ are zero mean Gaussian distributed random vectors whose means

and variances are as follows:

$$\mathbf{o}(k) \sim \aleph\left(\mathbf{0}, \begin{bmatrix} 0.15^2 & 0 & 0 & 0 \\ 0 & 0.4^2 & 0 & 0 \\ 0 & 0 & 0.08^2 & 0 \\ 0 & 0 & 0 & 0.15^2 \end{bmatrix}\right)$$

$$\mathbf{v}(k) \sim \aleph\left(\mathbf{0}, \begin{bmatrix} 0.04^2 & 0 & 0 & 0 \\ 0 & 0.04^2 & 0 & 0 \\ 0 & 0 & 0.04^2 & 0 \\ 0 & 0 & 0 & 0.04^2 \end{bmatrix}\right)$$

$$\mathbf{p}(k) \sim \aleph\left(\mathbf{0}, \begin{bmatrix} 0.15^2 & 0 \\ 0 & 0.17^2 \end{bmatrix}\right)$$

An initial block containing the first 300 input and output data points is selected as training data. Multi channel lattice filter is applied to this block for the generation of the residual vector $\boldsymbol{\varepsilon}_{1,\eta}(k)$. Since the process has four inputs and four outputs, $M = 5$ and $\eta = 1 \cdots 5(n+1) - 1$.

The initial process order n, which is two, is determined using the first 100 data points in the block. Then, the initial value of the covariance matrix, $\mathbf{R}_n^e(Int(\frac{300}{\tau})\tau)$ with $\tau = 3$, is estimated using $Int(\frac{300}{\tau}) = 100$ uncorrelated residual vectors.

Two scenarios of parameter variation are considered. First, starting at the time instant $k = 1501$ until $k = 6000$, the parameters of the following state matrix

$$\mathbf{A} = \begin{bmatrix} 0.67 & 0.67 \\ -0.67 & 0.67 \end{bmatrix}$$

are continuously changed at each time instant. For example, at the k^{th} time instant and thereafter at each time step, the first, second, and fourth elements decrease by 0.01%, but the third element increases by 0.01% of their respective values at the $(k-1)^{th}$ time instant. Consequently, at the time instant $k = 6000$,

$$\mathbf{A} = \begin{bmatrix} 0.4272 & 0.4272 \\ -0.4272 & 0.4272 \end{bmatrix}$$

Such a parametric variation is considered as a normal process drift because its magnitude is tiny.

Secondly, at $k = 8001$ time instant (No parametric changes are made during the period of time from $k = 6000$ until $k = 8000$.), the order of the system is changed from two to three, and at the same time additional parameters and process disturbance are introduced to the system. For instance, after such a change, matrices $\mathbf{A}, \mathbf{C}$ and the process disturbance $\mathbf{p}(k)$ are as follows

$$\mathbf{A} = \begin{bmatrix} 0.4272 & 0.4272 & -a \\ -0.4272 & 0.4272 & a \\ a & -a & a \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} -0.4326 & 0.1253 & -1.1456 & 1.1892 \\ -1.6656 & 0.2877 & 1.1909 & -0.0376 \\ b_1 & b_2 & b_3 & b_4 \end{bmatrix}$$

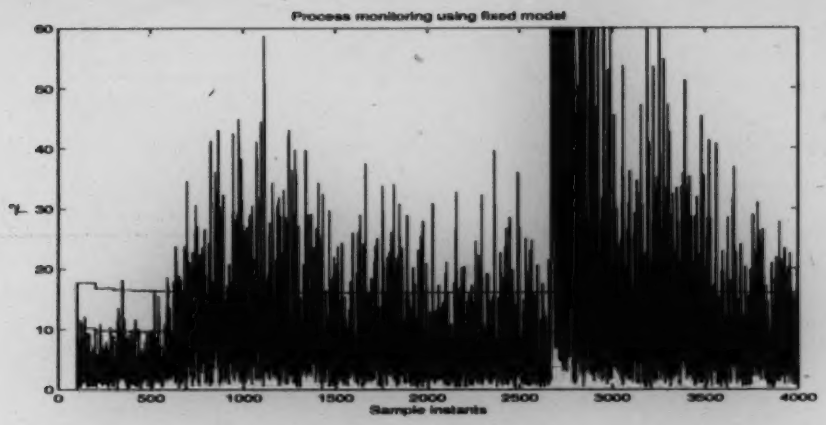


Figure 3.4: T^2 statistic for the Constant Model-based Process Monitoring on the Simulated System

$$\mathbf{C} = \begin{bmatrix} 0.3273 & -0.5883 & 0.2225 \\ 0.1746 & 2.1832 & -0.4532 \\ -0.1867 & -0.1364 & 0.2132 \\ 0.7258 & 0.1139 & 0.5674 \end{bmatrix}$$

$$\mathbf{p}(k) \sim \aleph\left(\mathbf{0}, \begin{bmatrix} 0.15^2 & 0 & 0 \\ 0 & 0.17^2 & 0 \\ 0 & 0 & 0.08^2 \end{bmatrix}\right)$$

where $a = 0.44$, $b_1 = -1.2332$, $b_2 = 0.4532$, $b_3 = 0.0987$, and $b_4 = -0.9987$.

Two approaches are used to monitor the above-mentioned time and order-varying process simultaneously, and the associated results are shown in Figures 3.4 and 3.5, respectively. Note that in both figures, (1) 95% and 99% confidence limits for $T^2(k)$ are selected; (2) the covariance matrix is updated based on $w_l = 100$ uncorrelated residual vectors if no alarm is announced; and (3) after every $n_{k+1} = 100$ samples, an on-line order re-determination is carried out.

In Figure 3.4, the residual vector $\mathbf{e}_n(k)$ is generated according to Equation 3.10 based on the known state space model matrices $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$. Since the model is constant, the calculated $T^2(k)$ frequently exceeds its confidence limit after the time instant $k = 1500$ even though the process is operating normally, giving a lot of false alarms. Such false alarms would not be tolerated in industry.

Figure 3.5 shows the $T^2(k)$ calculated from the residual vector generated by the adaptive multi channel lattice filter with a forgetting factor $\lambda = 0.99$. Since the lattice filter can adapt to the process variation, the T^2 statistic is basically within its 99% confidence limit for $k \leq 6000$, significantly eliminating false alarms. Further, when the order of the system changes at the time instant $k = 8001$, the new order is determined at the time instant $k = 8100$ using the index given by Equation 3.62. As shown clearly in the figure, the lattice filter-based $T^2(k)$ triggers its control limit during the transient period of time and drops

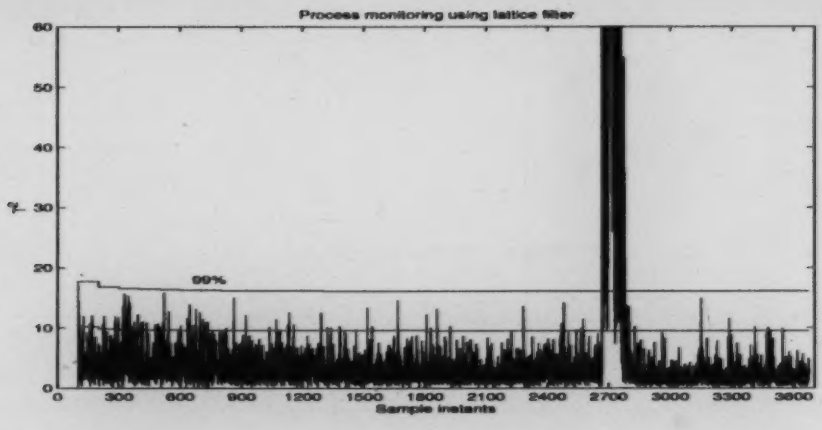


Figure 3.5: T^2 statistic for the Adaptive Lattice Filter-based Process Monitoring on the Simulated System

back to the limit after the system enters its new steady state, tracking the operational state of the system effectively.

Finally, note that on the x-axis of Figures 3.4 and 3.5, each point is not equal to each sample of the original data. Since the computed $T^2(k)$ is based on a sequence of uncorrelated residual vectors, before (including) point 2700, each point in the x-axis corresponds to three data samples ($\tau = 3$), and after point 2700, each point corresponds to four data samples ($\tau = 4$).

Also the parameters of the AR part in the ARMAX process are calculated to demonstrate the effective tracking by lattice filter. It is seen from Equation 3.66 that the initial values of the AR parameters are $\alpha_1 = -1.34$ and $\alpha_2 = 0.8978$, respectively. When there is a change in the parameters of the **A** matrix, the two parameters are also subject to a slight variation, e.g., at the instant $k = 6000$, $\alpha_1 = -0.8544$ and $\alpha_2 = 0.3650$. Moreover, at the instant $k = 8001$, when the order of the simulated process is changed from 2 to 3 suddenly, besides an extra parameter $\alpha_3 = -0.326$ being introduced to the AR part, α_1 and α_2 become -1.294 and 0.3650, respectively. Figure 3.6 plots the ratios of the estimated α_1, α_2 and α_3 over their respective true values. It can be deduced that the slowly time-varying parameters α_1 and α_2, and α_3 are effectively tracked.

3.7.2 Case Study Two- Pilot scale plant

In this study, the proposed scheme is used to monitor a real continuous stirred tank heater system (CSTHS). The system is located in the Computer Process Control Laboratory, in the Department of Chemical and Materials Engineering, University of Alberta, Canada. A schematic diagram of the CSTHS is given in Figure 3.7, where the cold water continuously flowing through the tank is heated by high temperature steam passing through a coil and four thermocouples (e.g., TT1, TT2, TT3 and TT4 in Figure 3.7) installed at different

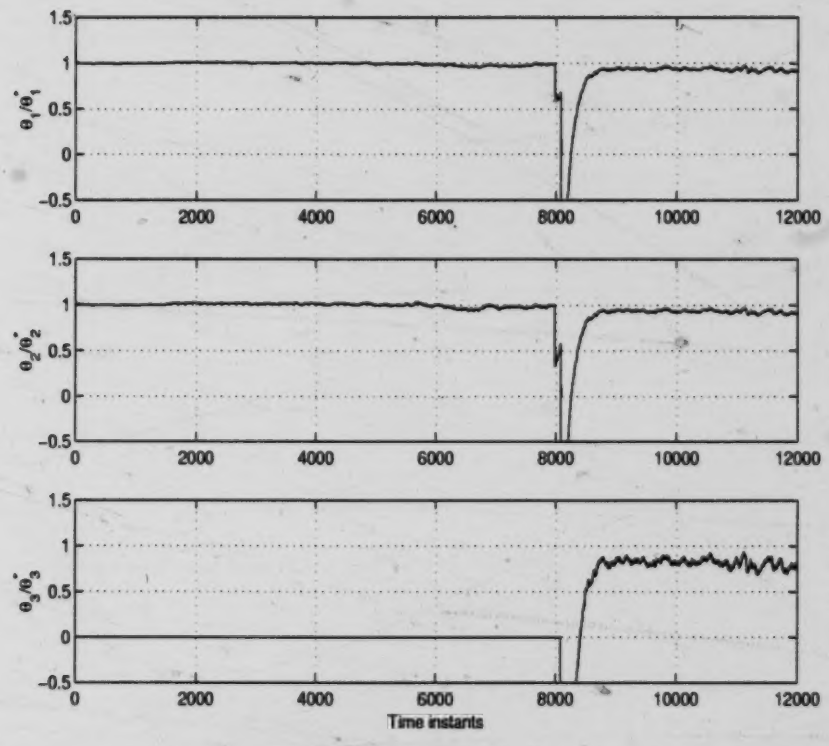


Figure 3.6: Tracking the Parameters of the AR Part of the Simulated System

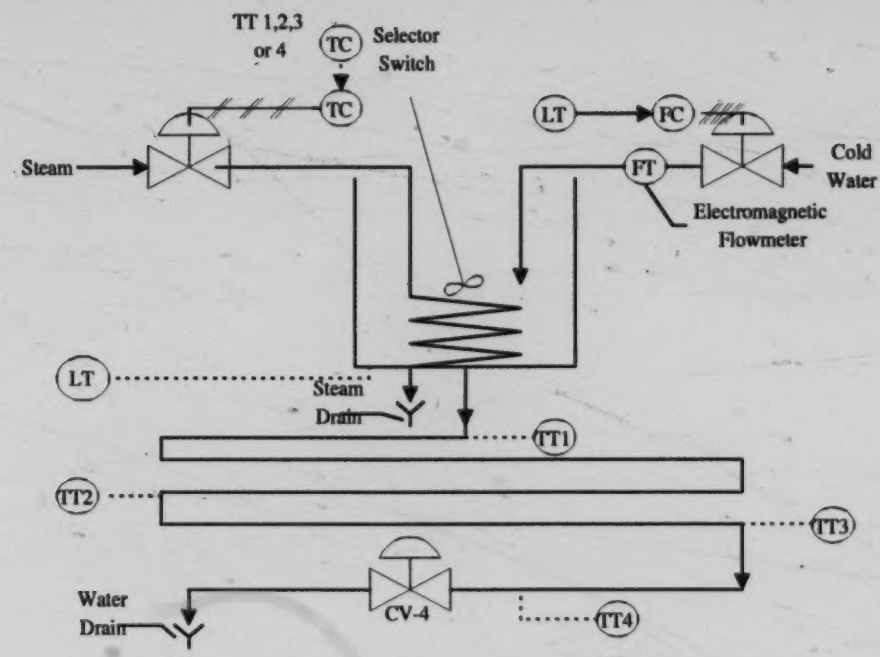


Figure 3.7: Schematic diagram of the CSTHS

locations of the long exit pipe provide temperature signals. The ultimate purpose of the CSTHS is to control the level and temperature of the water in the tank. There are three PID controllers included in the system. Two of them control the level and the temperature of the water in the tank respectively, and the third one controls the cold water flow rate.

In Figure 3.8, the lower subplot illustrates the flow rates of steam and cold water. As mentioned above, these are the manipulated variables for controlling temperature and level of the water in the tank. The upper subplot shows the corresponding level and temperature of the cold water.

The purpose of this case study is twofold. First, we would like to show again that our proposed adaptive monitoring scheme will not cause any false alarms when applied to a real pilot plant with a normal drift. Then, we show that in the event of a real fault, the proposed scheme can immediately detect the fault.

First 300 samples are collected from the CSTHS as the initial data block. Since the lattice filter converges very quickly, only the first 100 samples are used in the block to identify the order and coefficients of the lattice filter. The result of order determination is illustrated in Figure 3.11, where apparently the order is selected to be three based on the previous analysis. Further, the initial value of the covariance matrix of the residual vectors is estimated based on the whole block of data.

After time instant $k = 1500$, a slight drift is introduced into the system by adding a

small ramp to the level output, which is fed back to the level PID controller. Then, after time instant $k = 2000$, the setpoint of the water level is increased several times, after every 400 sampling intervals, by a magnitude 0.25 each time. Further, the setpoint is changed with a magnitude 0.40 after the time instant 3600. These setpoint changes in the level result in significant overall changes to the process. For example, when the level changes each time, the temperature to steam loop gain and time constant also change, making the CSTHS a time-varying system. However, it is required that these setpoint changes and a slight process drift should be considered as normal and not process faults from an engineering point of view.

As in the previous case study, two schemes are used to monitor the CSTHS simultaneously, where again, (1) uncorrelated residual vectors generated from the lattice filter are used to compute $T^2(k)$; (2) $w_l = 100$ uncorrelated residual vectors are used to update the covariance matrix; and (3) $n_{k+1} = 100$ residual vectors are employed to on-line re-determine the order at one time. In the first scheme, the lattice filter identified from the 100 samples in the initial block is used as a fixed model to generate the residual vector, and the corresponding result is given in Figure 3.9. Since the residuals are produced from the constant model, when the CSTHS is subjected to a normal variation, the computed $T^2(k)$ exceeds its confidence limit, especially its 95% confidence limit, frequently. These limit violations would clearly sound many false alarms.

Figure 3.10 depicts the result of multi channel lattice filter-based process monitoring and fault detection, where as compared with Figure 3.9, the number of false alarms are greatly reduced. Also between $k = 5200$ and $k = 5260$, a real fault is introduced into the CSTHS by choking the cold water exit pipe, and this fault is instantly detected by the statistic $T^2(k)$.

3.8 Conclusions

A multi-channel lattice filter scheme has been proposed for adaptive process monitoring of fully dynamic and time-varying processes.

The proposed scheme had been successfully applied to two case studies, a simulated process and a real pilot plant. In comparison with a constant model-based process monitoring technique, our proposed approach can adapt to a normal process drift, with significant reduction in the number of false alarms and yet at the same time is sensitive to any real faults. Also the proposed scheme can effectively track the slowly varying process parameters.

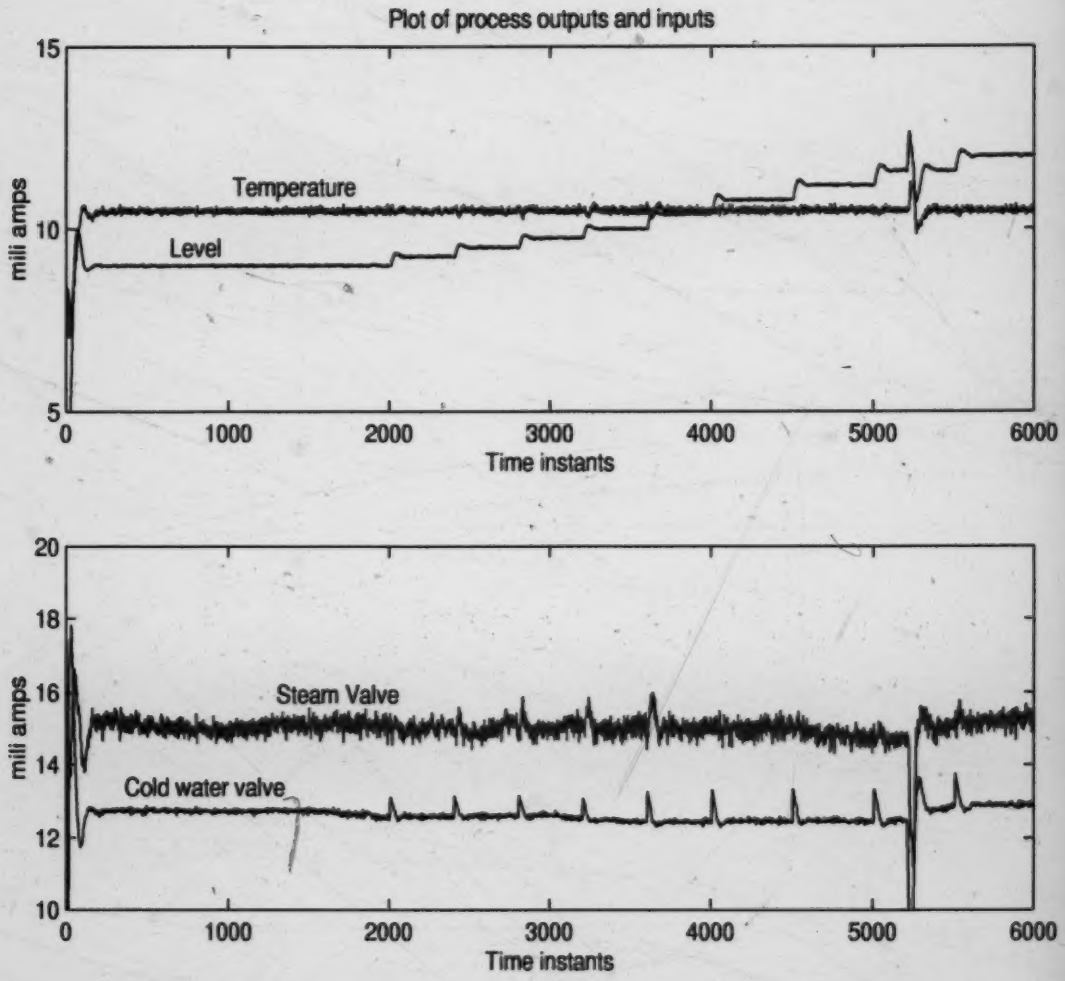


Figure 3.8: Flow Rates of Steam and Cold Water; Level and Temperature of Cold Water in CSTHS

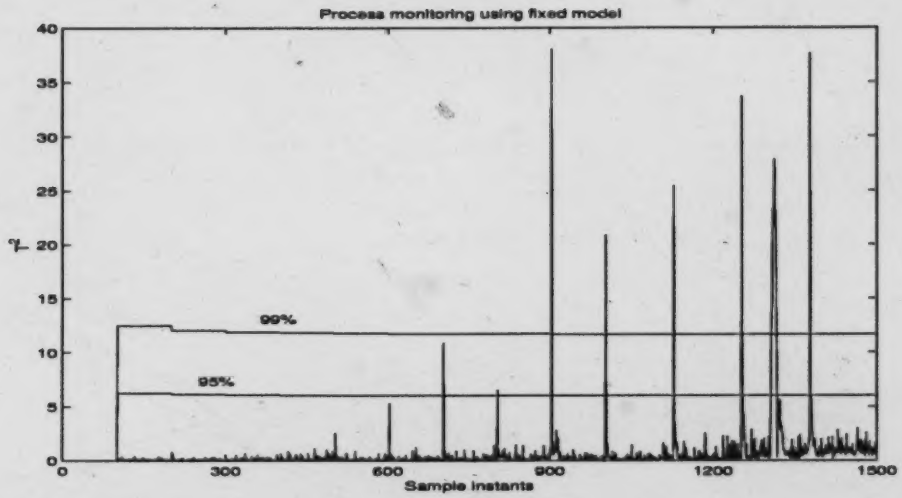


Figure 3.9: T^2 statistic for the Constant Model-based Process Monitoring on CSTHS (using every $4th$ sample period)

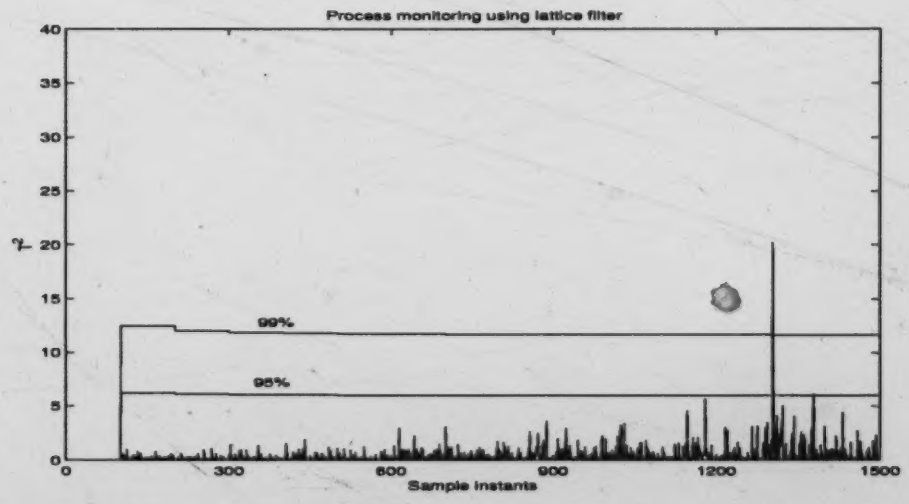


Figure 3.10: T^2 statistic for the Adaptive Lattice Filter-based Process Monitoring on CSTHS (using every $4th$ sample period)

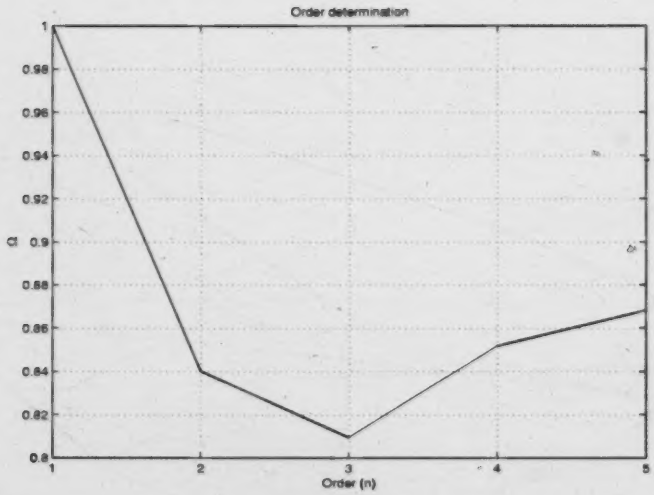


Figure 3.11: Order Determination of the CSTHS Model

Chapter 4

Sensor validation

4.1 Introduction

The detection and diagnosis of faults in complex process plants is an important task in the manufacturing and process industry. A process fault is understood as any kind of malfunction in the actual dynamic system or the plant that leads to an unacceptable anomaly in the overall system performance (Frank 1990). Such malfunctions may either occur in the sensors (instruments), or actuators, or other components of the process. It is desired that such faults be detected as early as possible and the root cause of the fault be identified before it evolves into a catastrophic event. In terms of the occurrence of the faults, they can basically be classified into *additive faults* and *multiplicative faults* (Gertler 1988). Sensor and actuator faults are the typical additive faults as they do not affect the model of the process under consideration. Process faults are typical multiplicative faults as they affect the process model parameters. The task of sensor fault detection, identification and reconstruction is referred as *sensor validation*.

The work in dynamic sensor validation falls into two categories: gross error detection in data reconciliation and sensor fault detection. Mah et al. (1976), Stanley and Mah (1977, 1981), and Romagnoli and Stephanopoulos (1981) are among the early researchers in gross error detection and rectification in chemical processes. More recent work provides advances in the use of dynamic models, nonlinear models, neural networks and additional statistical tests for the gross error detection (Crowe 1996). The work by Tong and Crowe (1995) is among the first to apply PCA to analysis of the model residuals. Dynamic gross error detection has also been studied by many researchers (Albuquerque and Biegler 1996, Karjala and Himmelblau 1996, Liebman *et al.* 1992). Kramer (1991) proposed the use of auto-associative neural networks for sensor fault detection, identification and reconstruction. Statistical process control (SPC) approaches, e.g., contribution plots (Miller *et al.* 1993) and dynamic principal component analysis techniques (Ku *et al.* 1995, Yoon and MacGregor 1998) have also been applied to sensor validation.

Deckert et al. (1977) have applied redundant hardware sensors to detect and identify abnormal sensors in the F-8 airplane . Other methods for sensor fault detection and isolation

(FDI) include the use of parity equations models (Chow and Willsky 1984), the parity equation-based structured residuals (Gertler and Singer 1985, Gertler and Singer 1990, Gertler 1991, Gertler and Kunwer 1995) and the use of observers and kalman filters (Clark 1979, Frank and Wunnenberg 1989). In addition, Basseville, based on the Chow-Willsky scheme, proposed a unified framework to handle the fault detection and isolation for a class of additive faults which can also be used for the detection and isolation of sensor faults (Basseville 1997, Basseville 1998).

Recently a new optimal method, *dynamic structured residual approach with maximized sensitivity* (DSRAMS), for the detection and identification of faulty sensors in fully dynamic systems has been proposed by Qin and Li (2000). The original concept of structured residuals has been the work of Gertler and his coworkers (Gertler and Singer, 1985, 1990). DSRAMS makes use of a normal process subspace model which can be identified using the EIV subspace identification algorithms. This method can also be directly applied to other dynamic models built either by the first principles or via system identification techniques such as state space, FIR, ARMAX, ARX, dynamic PCA or PLS models. Residuals, which are the linear combination of output and input variables, are generated making use of the process model for process monitoring. In order to reduce the rate of false alarms due to noise, an EWMA (exponentially weighted moving average) filter is applied to the residuals. For the identification of faulty sensors, DSRAMS is used to generate a set of structured residuals. All the structured residuals are then subjected to EWMA filter to reduce the effect of noise. For the identification of faulty sensors and for a reliable decision, the following three indices, the filtered structured residuals (FSRs), the cumulative sum (Qsum) and the cumulative variance (Vsum) are used. After the faulty sensor has been identified, the magnitude of the fault is estimated based on the model and the faulty data and the faulty sensor is reconstructed.

The subject matter of this chapter is organized as follows. The chapter starts with the model and fault representation in Section 2 which also deals with the derivation of fault detection index. Details of DSRAMS are outlined in section 3. It also discusses the conditions for detectability, attainability and isolability of sensor faults. Details of fault identification indices is carried out in Section 4 and the reconstruction of faulty sensor is presented in Section 5. Section 6 deals with the application of DSRAMS on a simulated example and a pilot scale plant. DSRAMS is evaluated experimentally by an on-line application to a pilot scale plant for the detection, identification and reconstruction of a single sensor fault. DSRAMS is also applied to a simulated system consisting of 5 outputs and 2 inputs where four different types of sensor faults: bias, drift, complete failure and precision degradation are introduced. The above method is then used for the identification of single and simultaneous multiple (two) sensor faults.

4.2 Residual Generation and Fault Detection Index

The section starts with the representation of a dynamic system by an EIV state space model which is followed by sensor fault representation. As will be seen, for residual generation only a subspace model needs to be identified. The section ends with the derivation of filtered fault detection index.

4.2.1 Fault Representation and Residual Generation

Model Representation

Again, as in Chapter 3, the dynamic process is represented by the state-space system:

$$\begin{aligned} \mathbf{x}(t+1) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\tilde{\mathbf{u}}(t) + \mathbf{p}(t) \\ \tilde{\mathbf{y}}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\tilde{\mathbf{u}}(t) \end{aligned} \tag{4.1}$$

where $\tilde{\mathbf{u}}(t) \in \Re^l$ and $\tilde{\mathbf{y}}(t) \in \Re^m$ are noise-free inputs and outputs respectively, and $\mathbf{x}(t) \in \Re^n$ is the state vector. In addition, $\mathbf{p}(t)$ is the process noise, and $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and $\mathbf{D}$ are system matrices with appropriate dimensions.

The observed inputs $\mathbf{u}(t)$ and outputs $\mathbf{y}(t)$ are assumed to be corrupted by zero-mean Gaussian distributed noise vectors $\mathbf{v}(t)$ and $\mathbf{o}(t)$ respectively, i.e.

$$\begin{aligned} \mathbf{u}(t) &= \tilde{\mathbf{u}}(t) + \mathbf{v}(t) \\ \mathbf{y}(t) &= \tilde{\mathbf{y}}(t) + \mathbf{o}(t) \end{aligned} \tag{4.2}$$

Substituting Equation 4.2 into Equation 4.1 yields

$$\begin{aligned} \mathbf{x}(t+1) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) - \mathbf{B}\mathbf{v}(t) + \mathbf{p}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) - \mathbf{D}\mathbf{v}(t) + \mathbf{o}(t) \end{aligned} \tag{4.3}$$

By performing algebraic manipulation in Equation 4.3, the following equation is obtained

$$\mathbf{y}_s(t) = \mathbf{\Gamma}_s \mathbf{x}(t-s) + \mathbf{H}_s \mathbf{u}_s(t) - \mathbf{H}_s \mathbf{v}_s(t) + \mathbf{G}_s \mathbf{p}_s(t) + \mathbf{o}_s(t) \tag{4.4}$$

where

$$\mathbf{y}_s(t) = \begin{bmatrix} \mathbf{y}(t-s) \\ \vdots \\ \mathbf{y}(t) \end{bmatrix} \in \Re^{m_s}$$

is the augmented output vector,

$$\mathbf{\Gamma}_s = \begin{bmatrix} \mathbf{C} \\ \vdots \\ \mathbf{C}\mathbf{A}^s \end{bmatrix} \in \Re^{m_s \times n}$$

is the extended observability matrix with rank n,

$$\mathbf{H}_s = \begin{bmatrix} \mathbf{D} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{C}\mathbf{B} & \mathbf{D} & \cdots & \mathbf{0} \\ \vdots & & \ddots & \vdots \\ \mathbf{C}\mathbf{A}^{s-1}\mathbf{B} & & \cdots & \mathbf{D} \end{bmatrix} \in \Re^{m_s \times l_s}$$

and

$$\mathbf{G}_s = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{C} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & & \ddots & \vdots \\ \mathbf{CA}^{s-1} & & \cdots & \mathbf{0} \end{bmatrix} \in \Re^{m_s \times n_s}$$

are two lower triangular block Toeplitz matrices with $m_s = (s+1)m$, $l_s = (s+1)l$ and $n_s = (s+1)n$. In fact $\mathbf{H}_s$ is the Hankel matrix with the impulse response coefficients as its elements. The vectors $\mathbf{v}_s(t) \in \Re^{l_s}$, $\mathbf{o}_s(t) \in \Re^{m_s}$, $\mathbf{p}_s(t) \in \Re^{n_s}$, and $\mathbf{u}_s(t) \in \Re^{l_s}$ are defined in the same manner as $\mathbf{y}_s(t)$. Also it is to note that s is the extended observability index which should be selected at least to be equal to the process order n (Ding *et al.* 1999).

In addition, define

$$\mathbf{z}_s(t) = \begin{bmatrix} \mathbf{y}_s(t) \\ \mathbf{u}_s(t) \end{bmatrix} \in \Re^{m_s + l_s}$$

and

$$\tilde{\mathbf{H}}_s = [\mathbf{I}_{m_s}| - \mathbf{H}_s] \in \Re^{m_s \times (m_s + l_s)}$$

where $\mathbf{I}_{m_s} \in \Re^{m_s \times m_s}$ is an identity matrix. Consequently, Equation 4.4 can be rewritten as

$$\tilde{\mathbf{H}}_s \mathbf{z}_s(t) = \mathbf{\Gamma}_s \mathbf{x}(t-s) - \mathbf{H}_s \mathbf{v}_s(t) + \mathbf{G}_s \mathbf{p}_s(t) + \mathbf{o}_s(t) \tag{4.5}$$

Fault Representation

Now if a sensor becomes faulty, then its measurement will contain the normal value of the measured process variables and the unknown fault magnitude . Therefore, in the presence of sensor faults, inputs and outputs can be represented as follows:

$$\begin{array}{rcl} \mathbf{u}(t) & = & \mathbf{u}^*(t) + \mathbf{\Xi}_{u,j}\mathbf{f}_{u,j}(t) \\ \mathbf{y}(t) & = & \mathbf{y}^*(t) + \mathbf{\Xi}_{y,i}\mathbf{f}_{y,i}(t) \end{array} \tag{4.6}$$

where $\mathbf{u}^*(t), \mathbf{y}^*(t)$ are fault free input and output values respectively, $\mathbf{\Xi}_{u,j} \in \Re^{l \times l_j}$ and $\mathbf{\Xi}_{y,i} \in \Re^{m \times l_i}$ are matrices of fault directions, and $\mathbf{f}_{u,j}(t) \in \Re^{l_j}$ and $\mathbf{f}_{y,i}(t) \in \Re^{l_i}$ are fault magnitude vectors. To represent a single sensor fault in the i^{th} output sensor, for $i = 1, \cdots, m$, $\mathbf{\Xi}_{y,i} = [0\,0 \cdots 1 \cdots 0]^T \in \Re^m$, which is the i^{th} column of the identity matrix $\mathbf{I}_m$. In the presence of simultaneous multiple sensor faults, $\mathbf{\Xi}_{y,i}$ will simply consist of the corresponding columns of the identity matrix. For instance, consider a dynamic system of five sensors of which sensors 1 and 3 become simultaneously faulty. In such a scenario,

$$\mathbf{\Xi}_{y,i} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \in \Re^{5 \times 2}$$

In terms of $\mathbf{z}_s(t)$, as defined earlier, the sensor fault can be represented as

$$\mathbf{z}_s(t) = \mathbf{z}_s^*(t) + \mathbf{\Xi}_{z,i}\mathbf{f}_{z,i}(t) \tag{4.7}$$

where

$$\mathbf{z}_s^*(t) = \begin{bmatrix} \mathbf{y}_s^*(t) \\ \mathbf{u}_s^*(t) \end{bmatrix}$$

$$\Xi_{z,i} = \begin{bmatrix} \Xi_{y,i} & 0 & \cdots & 0 \\ 0 & \ddots & & \\ \vdots & & \ddots & \\ 0 & & & \Xi_{y,i} \\ 0 & \cdots & & 0 \\ \vdots & & & \\ 0 & \cdots & & 0 \end{bmatrix} \in \Re^{(m_s+l_s)\times(s+1)l_i}$$

$$\mathbf{f}_{z,i}(t) = \left[\mathbf{f}_{y,i}^T(t-s), \cdots, \mathbf{f}_{y,i}^T(t)\right]^T \in \Re^{(s+1)l_i}$$

for output sensor faults

$$\Xi_{z,i} = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & & \\ 0 & \cdots & 0 \\ \Xi_{u,i} & 0 & \\ 0 & \ddots & \\ \vdots & \ddots & \\ 0 & & \Xi_{u,i} \end{bmatrix} \in \Re^{(m_s+l_s)\times(s+1)l_i}$$

$$\mathbf{f}_{z,i}(t) = \left[\mathbf{f}_{u,i}^T(t-s), \cdots, \mathbf{f}_{u,i}^T(t)\right]^T \in \Re^{(s+1)l_i}$$

for actuator or input sensor faults and

$$\Xi_{z,i} = \begin{bmatrix} \Xi_{y,i} & 0 & \cdots & 0 & 0 & \cdots & & 0 \\ 0 & \ddots & & & \vdots & & & \vdots \\ \vdots & & \ddots & & \vdots & & & \vdots \\ 0 & & & \Xi_{y,i} & 0 & \cdots & & 0 \\ 0 & \cdots & & 0 & \Xi_{u,i} & 0 & \cdots & 0 \\ \vdots & & & \vdots & 0 & \ddots & & \\ \vdots & & & \vdots & \vdots & & \ddots & \\ 0 & \cdots & & 0 & 0 & & & \Xi_{u,i} \end{bmatrix} \in \Re^{(m_s+l_s)\times 2(s+1)l_i}$$

$$\mathbf{f}_{z,i}(t) = \left[\mathbf{f}_{y,i}^T(t-s), \cdots, \mathbf{f}_{y,i}^T(t), \mathbf{f}_{u,i}^T(t-s), \cdots, \mathbf{f}_{u,i}^T(t)\right]^T \in \Re^{2(s+1)l_i}$$

for simultaneous output and input sensor faults.

Residual Generation

Substituting Equation 4.7 in Equation 4.5 gives

$$\begin{aligned}\tilde{\mathbf{H}}_s\mathbf{z}_s(t) &= \tilde{\mathbf{H}}_s\mathbf{z}_s^*(t) + \tilde{\mathbf{H}}_s\boldsymbol{\Xi}_{z,i}\mathbf{f}_{z,i}(t) \\ &= \boldsymbol{\Gamma}_s\mathbf{x}(t-s) - \mathbf{H}_s\mathbf{v}_s(t) + \mathbf{G}_s\mathbf{p}_s(t) + \mathbf{o}_s(t) + \tilde{\mathbf{H}}_s\boldsymbol{\Xi}_{z,i}\mathbf{f}_{z,i}(t) \qquad (4.8)\end{aligned}$$

The residual vector is generated by eliminating the state vector, $\mathbf{x}(t-s)$, from the above equation. This is achieved by pre-multiplying the above equation by the orthogonal complement of the extended observability matrix $\boldsymbol{\Gamma}_s$. As the range subspace of $\boldsymbol{\Gamma}_s$ is n-dimensional, its orthogonal complement is $(m_s - n)$-dimensional.

Denoting the matrix $\boldsymbol{\Gamma}_s^{\perp} \in \Re^{(m_s-n)\times m_s}$ to be the orthogonal complement of $\boldsymbol{\Gamma}_s$, the primary residual vector is generated as follows

$$\mathbf{e}(t) \equiv \boldsymbol{\Gamma}_s^{\perp}\tilde{\mathbf{H}}_s\mathbf{z}_s(t) \in \Re^{(m_s-n)} \qquad (4.9)$$

Pre-multiplying Equation 4.8 by the matrix $\boldsymbol{\Gamma}_s^{\perp}$ leads to

$$\begin{aligned}\mathbf{e}(t) &= -\boldsymbol{\Gamma}_s^{\perp}\mathbf{H}_s\mathbf{v}_s(t) + \boldsymbol{\Gamma}_s^{\perp}\mathbf{G}_s\mathbf{p}_s(t) + \boldsymbol{\Gamma}_s^{\perp}\mathbf{o}_s(t) + \boldsymbol{\Gamma}_s^{\perp}\tilde{\mathbf{H}}_s\boldsymbol{\Xi}_{z,i}\mathbf{f}_{z,i}(t) \\ &= \mathbf{e}^*(t) + \boldsymbol{\Gamma}_s^{\perp}\tilde{\mathbf{H}}_s\boldsymbol{\Xi}_{z,i}\mathbf{f}_{z,i}(t) \qquad (4.10)\end{aligned}$$

where

$$\mathbf{e}^*(t) = -\boldsymbol{\Gamma}_s^{\perp}\mathbf{H}_s\mathbf{v}_s(t) + \boldsymbol{\Gamma}_s^{\perp}\mathbf{G}_s\mathbf{p}_s(t) + \boldsymbol{\Gamma}_s^{\perp}\mathbf{o}_s(t)$$

In the absence of faults, $\mathbf{e}(t) = \mathbf{e}^*(t)$ is a zero-mean noise vector. However, in the presence of faults, the mean of $\mathbf{e}(t)$ changes to $\boldsymbol{\Gamma}_s^{\perp}\tilde{\mathbf{H}}_s\boldsymbol{\Xi}_{z,i}\mathbf{f}_{z,i}(t)$. Therefore, by observing the mean of $\mathbf{e}(t)$, any faults can be detected.

4.2.2 Fault Detection Index

From Equation 4.10, in the absence of faults,

$$\mathbf{e}(t) = \mathbf{e}^*(t)$$

is a moving average (MA) of the noise vectors $\mathbf{v}(t), \mathbf{p}(t)$ and $\mathbf{o}(t)$. Under the assumption that $\mathbf{v}(t), \mathbf{p}(t)$ and $\mathbf{o}(t)$ are zero-mean Gaussian distributed, it follows that

$$\mathbf{e}^*(t) \sim \aleph(0, \mathbf{R}_e)$$

or equivalently,

$$\mathbf{e}^{*T}(t)\mathbf{R}_e^{-1}\mathbf{e}^*(t) \sim \chi^2(m_s - n)$$

where $\mathbf{R}_e$ is the covariance matrix of $\mathbf{e}^*(t)$ which can be estimated from the normal process data.

The fault detection index is defined as

$$d(t) = \mathbf{e}^T(t)\mathbf{R}_e^{-1}\mathbf{e}(t)$$

and its confidence limit is $d_\alpha = \chi^2_\alpha(m_s - n)$, where α is a pre-selected level of significance. The presence of a fault is indicated whenever the fault detection index, $d(t)$, exceeds its confidence limit.

Filtered Detection Index

If the process data is too noisy, then the use of $d(t)$ as the fault detection index might result in a number of false alarms. To overcome this problem, an exponentially weighted moving average (EWMA) filter is applied to $\mathbf{e}(t)$ and the filtered residual vector is

$$\bar{\mathbf{e}}(t) = \gamma\bar{\mathbf{e}}(t-1) + (1-\gamma)\mathbf{e}(t)$$

where $0 \leq \gamma < 1$. Hence the filtered fault detection index

$$\bar{d}(t) = \bar{\mathbf{e}}^T(t)\bar{\mathbf{R}}_e^{-1}\bar{\mathbf{e}}(t)$$

is used for monitoring the process where $\bar{\mathbf{R}}_e$ is the covariance matrix of $\bar{\mathbf{e}}(t)$. It should be noted that $\bar{\mathbf{e}}(t)$ is no longer an uncorrelated time series. However, if we assume that $\bar{\mathbf{e}}(t)$ is still gaussian distributed then $\bar{d}(t)$ is still χ^2 distributed with a reduced degree of freedom governed by the rank of the covariance matrix. If $\bar{\mathbf{R}}_e$ is ill-conditioned, indicating highly time correlated measurements, then one can perform Singular Value Decomposition on the residuals and work with the reduced transformed set of residuals.

4.3 Fault Identification with DSRAMS

After the detection of faults by the fault detection index, the root cause of the faults need to be investigated. This is referred to as *fault isolation* (FI). There are a number of approaches available for fault isolation. This section deals with one such approach called *dynamic structured residual approach with maximized sensitivity* (DSRAMS).

4.3.1 Concept of Structured Residuals

The primary residual vector, $\mathbf{e}(t)$, can only be used for the detection of faults. For the isolation of faults, there is a need to transform $\mathbf{e}(t)$ into a new residual vector $\mathbf{r}(t)$, i.e., $\mathbf{r}(t) = \mathbf{W}\mathbf{e}(t)$. The residual vector $\mathbf{r}(t)$ is termed as the *structured residual vector* where one element is structured to be insensitive to a specified set of faults while sensitive to the others.

The design of structured residual vector is equivalent to the calculation of the transformation matrix $\mathbf{W}$. The structure of the transformation matrix, $\mathbf{W}$, is characterized by an incidence matrix which consists of binary code "1" and "0". The rows in the incidence matrix correspond to each element of $\mathbf{r}(t)$ while the columns correspond to faults in sensors or actuators. The procedure for the design of structured residual vector can be explained

by considering the following 2 input - 2 output second order dynamic system

$$\begin{aligned}\mathbf{e}(t) &= \mathbf{y}(t) + \mathbf{A}_1\mathbf{y}(t-1) + \mathbf{A}_2\mathbf{y}(t-2) - \mathbf{B}_0\mathbf{u}(t) - \mathbf{B}_1\mathbf{u}(t-1) - \mathbf{B}_2\mathbf{u}(t-2) \\ &= [\mathbf{A}_2 \;\; \mathbf{A}_1 \;\; \mathbf{A}_0 \;\; -\mathbf{B}_2 \;\; -\mathbf{B}_1 \;\; -\mathbf{B}_0] \begin{bmatrix} \mathbf{y}(t-2) \\ \mathbf{y}(t-1) \\ \mathbf{y}(t) \\ \mathbf{u}(t-2) \\ \mathbf{u}(t-1) \\ \mathbf{u}(t) \end{bmatrix}\end{aligned}$$

where $\mathbf{A}_0 = \mathbf{I}_2$ and $\mathbf{A}_i, \mathbf{B}_i$, $i = 0, 1, 2$ are the matrices consisting of the coefficients of output and input variables respectively. For the isolation of sensor or actuator faults, a structured residual vector $\mathbf{r}(t)$ containing four elements is designed according to the following incidence matrix

	$y_1(t-2)$	$y_2(t-2)$	$y_1(t-1)$	$y_2(t-1)$	$y_1(t)$	$y_2(t)$
$r_1(t)$	0	1	0	1	0	1
$r_2(t)$	1	0	1	0	1	0
$r_3(t)$	1	1	1	1	1	1
$r_4(t)$	1	1	1	1	1	1

	$u_1(t-2)$	$u_2(t-2)$	$u_1(t-1)$	$u_2(t-1)$	$u_1(t)$	$u_2(t)$
$r_1(t)$	1	1	1	1	1	1
$r_2(t)$	1	1	1	1	1	1
$r_3(t)$	0	1	0	1	0	1
$r_4(t)$	1	0	1	0	1	0

where "0" indicates an element in $\mathbf{r}(t)$ to be insensitive and "1" indicates an element that is sensitive to a fault. In the above incidence matrix, each element of $\mathbf{r}(t)$ is unaffected by one specified fault and its time shifted values. For instance, $r_1(t)$ is designed to be insensitive to $y_1(t)$, $y_1(t-1)$ and $y_1(t-2)$ but sensitive to the faults in others. The design of transformation matrix $\mathbf{W}$ to obtain the structured residual vector $\mathbf{r}(t)$ for fault isolation is based on such a incidence matrix, as above, and the residual model. Therefore, after the detection of a fault, by observing the response of $\mathbf{r}$(t) to fault via time, one can correctly pinpoint the faulty sensor or actuator. For instance, after the alarm, if $r_1(t)$ is unaffected but all other elements in $\mathbf{r}$(t) are affected by the fault, then it can be concluded that the sensor measuring $y_1(t)$ is faulty.

In the case of multiple sensor faults, the ith structured residual is designed to be insensitive to a group of sensors, g_i, where $i \in g_i$, and most sensitive to the faults in other sensors with index $\bar{g}_i$. For instance, for the case of four sensors, one can design the incidence vector for the first residual as [0 1 0 1], which makes the residual insensitive to faults in sensors 1 and 3. However one must ensure that, for the desirable incidence matrix, attainability and isolability conditions are satisfied.

4.3.2 Detectability, Attainability and Isolability

Detectability, attainability and isolability are important issues for any fault detection and isolation scheme. There are certain conditions that should be satisfied for the faults to be detectable, the selected structure for $\mathbf{r}(t)$ to be attainable and for any two faults to be isolable. In order to investigate the conditions for *Detectability, Attainability* and *Isolability*, the above example of 2×2 second order system is considered in the following subsections.

Detectability

After the occurrence of a fault, the residual vector is,

$$\mathbf{e}(t) = \mathbf{e}^*(t) + \begin{cases} [\mathbf{A}_2\mathbf{e}_i \quad \mathbf{A}_1\mathbf{e}_i \quad \mathbf{A}_0\mathbf{e}_i] \begin{bmatrix} \triangle y_i(t-2) \\ \triangle y_i(t-1) \\ \triangle y_i(t) \end{bmatrix} & \text{If } i^{th} \text{ sensor is faulty, } i = 1,2 \\ [-\mathbf{B}_2\mathbf{e}_i \quad -\mathbf{B}_1\mathbf{e}_i \quad -\mathbf{B}_0\mathbf{e}_i] \begin{bmatrix} \triangle u_i(t-2) \\ \triangle u_i(t-1) \\ \triangle u_i(t) \end{bmatrix} & \text{If } i^{th} \text{ actuator is faulty, } i = 1,2 \end{cases}$$

where $\mathbf{e}_i$ is the i^{th} column (i=1,2) of identity matrix $\mathbf{I}_2$. The sufficient condition for a fault to be detectable is that none of the rows of

$$[\mathbf{A}_2\mathbf{e}_i \quad \mathbf{A}_1\mathbf{e}_i \quad \mathbf{A}_0\mathbf{e}_i]$$

are orthogonal to the fault vector

$$\begin{bmatrix} \triangle y_i(t-2) \\ \triangle y_i(t-1) \\ \triangle y_i(t) \end{bmatrix}$$

or none of the rows of

$$[-\mathbf{B}_2\mathbf{e}_i \quad -\mathbf{B}_1\mathbf{e}_i \quad -\mathbf{B}_0\mathbf{e}_i]$$

are orthogonal to the fault vector

$$\begin{bmatrix} \triangle u_i(t-2) \\ \triangle u_i(t-1) \\ \triangle u_i(t) \end{bmatrix}$$

Hence detectability depends on the model of the process.

Attainability

The transformation matrix $\mathbf{W}$ should have a non-trivial solution for a given incidence matrix. For instance, if $r_1(t) = \mathbf{w}_1^T\mathbf{e}(t)$ is designed to be insensitive to $y_1(t)$, $y_1(t-1)$ and $y_1(t-2)$, where $\mathbf{w}_1^T \in \Re^{1\times 2}$ is the first row of matrix $\mathbf{W}$, then geometrically, $\mathbf{w}_1$ should be orthogonal to the columns of $[\mathbf{A}_2\mathbf{e}_i \quad \mathbf{A}_1\mathbf{e}_i \quad \mathbf{A}_0\mathbf{e}_i]$, i.e.,

$$\mathbf{w}_1^T [\mathbf{A}_2\mathbf{e}_i \quad \mathbf{A}_1\mathbf{e}_i \quad \mathbf{A}_0\mathbf{e}_i] = 0$$

A non-trivial solution to $\mathbf{w}_1^T$ exists only if rank($[\mathbf{A}_2\mathbf{e}_i \;\; \mathbf{A}_1\mathbf{e}_i \;\; \mathbf{A}_0\mathbf{e}_i]$) = 1, otherwise $\mathbf{w}_1^T$ only has a trivial solution. Similar conditions apply for the other rows of $\mathbf{W}$. The selected structure for $\mathbf{r}(t)$ is said to be *attainable* only if a non-trivial solution to $\mathbf{W}$ exists.

The *attainability* condition for a selected incidence matrix is that the rank of columns in the residual model, e.g., $[\mathbf{A}_2\mathbf{e}_i \;\; \mathbf{A}_1\mathbf{e}_i \;\; \mathbf{A}_0\mathbf{e}_i]$, corresponding to the "0s" in the incidence matrix, should be less than the number of columns of $\mathbf{W}$. Therefore, to ensure a selected structure for $\mathbf{r}(t)$ to be attainable, the number of "0s" in each row of the incidence matrix should at most be

$$\textit{number of columns in } \mathbf{W} - 1$$

Isolability

After the computation of the transformation matrix, $\mathbf{W}$, the model for structured residual vector is

$$\mathbf{P} \equiv \mathbf{W}[\mathbf{A}_2 \;\; \mathbf{A}_1 \;\; \mathbf{A}_0 \;\; -\mathbf{B}_2 \;\; -\mathbf{B}_1 \;\; -\mathbf{B}_0]$$

If two or more columns in $\mathbf{P}$ are linearly dependent, then the faults corresponding to those columns are not isolable.

4.3.3 Dynamic Structured Residual Approach with Maximized Sensitivity (DSRAMS)

The determination of the structure for $\mathbf{r}(t)$ has been discussed in the previous sections. This section deals with the calculation of the parameters of transformation matrix, $\mathbf{W}$, for dynamic processes.

For multi-dimensional faults, the matrix, $\mathbf{\Gamma}_s^{\perp}\tilde{\mathbf{H}}_s\mathbf{\Xi}_{z,i}$ can have linearly dependent columns. Under such circumstances the sensor faults associated with the linearly related columns will not be isolable. Hence singular value decomposition (SVD) is used to remove the correlations between the columns of the matrix. Performing SVD on matrix $\mathbf{\Gamma}_s^{\perp}\tilde{\mathbf{H}}_s\mathbf{\Xi}_{z,i}$ results in

$$\mathbf{\Gamma}_s^{\perp}\tilde{\mathbf{H}}_s\mathbf{\Xi}_{z,i} = \mathbf{U}_i\mathbf{D}_i\mathbf{V}_i^T$$

where $\mathbf{U}_i \in \Re^{(m_s-n)\times l_{s,i}}, \mathbf{D}_i \in \Re^{l_{s,i}\times l_{s,i}}, \mathbf{V}_i \in \Re^{l_{s,i}\times l_{s,i}}$ and $l_{z,i}$ is the number of non-zero singular values of $\mathbf{\Gamma}_s^{\perp}\tilde{\mathbf{H}}_s\mathbf{\Gamma}_{z,i}$. Consequently, Equation 4.10 can be rewritten as

$$\begin{aligned}\mathbf{e}(t) &= \mathbf{e}^*(t) + \mathbf{U}_i\mathbf{D}_i\mathbf{V}_i^T\mathbf{f}_{z,i}(t) \\ &= \mathbf{e}^*(t) + \mathbf{U}_i\mathbf{f}_i(t) \qquad (4.11)\end{aligned}$$

where

$$\mathbf{f}_i(t) = \mathbf{D}_i\mathbf{V}_i^T\mathbf{f}_{z,i}(t)$$

is the magnitude of fault projected on $\mathbf{U}_i$.

For the identification of faulty sensor, a set of structured residuals are generated in which the i^{th} residual $r_i(t)$ is insensitive to the current and past values of $\mathbf{f}_{z,i}(t), \cdots, \mathbf{f}_{z,i}(t-s)$ but

most sensitive to the faults in other sensors. Making use of Equation 4.11, the i^{th} structured residual is generated as follows,

$$\begin{aligned} r_i(t) &= \mathbf{w}_i^T \mathbf{e}(t) \\ &= \mathbf{w}_i^T \mathbf{e}^*(t) + \mathbf{w}_i^T \mathbf{U}_i \mathbf{f}_i(t) \end{aligned} \tag{4.12}$$

where the vector $\mathbf{w}_i \in \Re^{m_s - n}$ should be orthogonal to the columns of $\mathbf{U}_i$ while minimizing its angles with the columns of the matrix $\mathbf{U}_j$ for $j \neq i$.

The above criterion of designing $\mathbf{w}_i$ for $i = 1, \cdots, m_f$, where m_f is the total number of faults, is equivalent to

$$\max_{\mathbf{w}_i} \sum_{j=1}^{m_f} \| \mathbf{U}_j^T \mathbf{w}_i \|^2 \tag{4.13}$$

subject to $\mathbf{U}_i^T \mathbf{w}_i = \mathbf{0}$ and $\| \mathbf{w}_i \| = 1$. Selecting $\mathbf{w}_i$ in the orthogonal complement of $\mathbf{U}_i$, i.e.,

$$\mathbf{w}_i = \left(\mathbf{I} - \mathbf{U}_i \mathbf{U}_i^T\right) \omega_i$$

Equation 4.13 is then re-phrased as

$$\max_{\omega_i} \sum_{j=1}^{m_f} \| \mathbf{U}_j^T \left(\mathbf{I} - \mathbf{U}_i \mathbf{U}_i^T\right) \omega_i \|^2 = \max_{\omega_i} \sum_{j=1}^{m_f} \| \mathbf{U}_{ji}^T \omega_i \|^2 \tag{4.14}$$

subject to

$$\| \left(\mathbf{I} - \mathbf{U}_i \mathbf{U}_i^T\right) \omega_i \| = 1$$

where $\mathbf{U}_{ji} = \left(\mathbf{I} - \mathbf{U}_i \mathbf{U}_i^T\right) \mathbf{U}_i$ is the projection of $\mathbf{U}_j$ on the orthogonal complement of $\mathbf{U}_i$. The solution to the above equation reduces to solving the following eigenvalue problem:

$$\sum_{j=1}^{m_f} \mathbf{U}_{ji} \mathbf{U}_{ji}^T \mathbf{w}_i = \lambda \mathbf{w}_i$$

The eigenvector of $\sum_{j=1}^{m_f} \mathbf{U}_{ji} \mathbf{U}_{ji}^T$ corresponding to the largest eigenvalue is the required vector $\mathbf{w}_i$. The largest eigenvalue only satisfies the sufficient condition for Equation 4.14 to achieve the maximum.

4.4 Fault Identification Indices

The following subsections define three types of fault identification indices: (1) EWMA -filtered structured residuals (FSR), (2) Cumulative sum (Q_{sum}) and (3) Cumulative variance (V_{sum}). The motive behind using more than one identification index is to make a reliable decision when dealing with different types of faults.

4.4.1 Filtered Structured Residuals (FSR)

In the absence of sensor failures, ideally, each structured residual should be equal to zero. However, due to uncertainties such as modeling errors, measurement noise etc., $r_i(t)$ is never equal to zero. So there is a need for determining non-zero confidence limit for each $r_i(t), i = 1, 2, \cdots, m_f$, using statistical techniques.

From Equation 4.11 it can be inferred that

$$r_i(t) = \mathbf{w}_i^T \mathbf{e}^*(t) \sim \aleph(0, \sigma_i^2), \quad i = 1, 2, \cdots, m_f.$$

and,

$$\frac{r_i^2(t)}{\sigma_i^2} \sim \chi^2(1)$$

where $\sigma_i^2 = E\left\{\mathbf{w}_i^T \mathbf{R}_e \mathbf{w}_i\right\}$. The confidence limit for $r_i^2(t)$ is $\eta_i^\alpha = \sigma_i^2 \chi_\alpha^2(1)$. When an EWMA filter is applied to $r_i(t)$, the filtered structured residual is as follows:

$$\bar{r}_i(t) = \gamma \bar{r}_i(t-1) + (1-\gamma) r_i(t)$$

and the threshold for $\bar{r}_i^2(t)$ is

$$\bar{\eta}_i^\alpha = \bar{\sigma}_i^2 \chi_\alpha^2(1), \ i = 1, \cdots, m_f.$$

where $\bar{\sigma}_i^2 = E\left\{\mathbf{w}_i^T \bar{\mathbf{R}}_e \mathbf{w}_i\right\}$. For better visualization an identification index based on filtered structured residual is then defined as follows

$$I_{FSR}^i(t) = \frac{r_i^2(t)}{\bar{\eta}_i^\alpha}$$

Under normal condition, $I_{FSR}^i(t)$ $(i = 1, \cdots, m + l)$ is less than one. However, if a sensor $i \in g_i$ is faulty, $I_{FSR}^i(t) < 1$ but all other $I_{FSR}^j(t) > 1$ $(j \in \bar{g}_i)$.

4.4.2 Cumulative Sum (Q_{sum})

The Cumulative sum index, which is based on the generalized likelihood ratio, is sensitive to sensor faults that incur significant changes in mean such as bias, drift and complete failure. As $r_i(t)$ is finitely autocorrelated, a sequence of uncorrelated signals $\{r_i(t_f + k\Delta)\}$ for $k = 0, 1, \cdots, t - t_f$, where t_f is the fault detection time, is used from the sequence $\{r_i(t_f), \cdots, r_i(t)\}$. The cumulative sum index, Q_{sum}, is defined as follows

$$Q_{sum}(t, i) = \sum_{k=0}^{\frac{t-t_f}{\Delta}} r_i(t_f + k\Delta)$$

A normalized index for Q_{sum} for the identification of faults is then defined as

$$I_{Q_{sum}}^i(t) = \frac{Q_{sum}^2(t, i)}{(\frac{t-t_f}{\Delta}) \mathbf{w}_i^T \mathbf{R}_e \mathbf{w}_i \chi_\alpha^2(1)}$$

4.4.3 Cumulative Variance (V_{sum})

The Cumulative variance index is sensitive to sensor faults that cause variance to change, e.g., a precision degradation fault. The V_{sum} index is calculated as follows

$$V_{sum}(t,i) = \sum_{k=0}^{\frac{t-t_f}{\Delta}} (r_i(t_f + k\Delta) - \mu_i)^2$$

where

$$\mu_i = \frac{\sum_{k=0}^{\frac{t-t_f}{\Delta}} r_i(t_f + k\Delta)}{1 + \frac{t-t_f}{\Delta}}$$

Again the normalized index for V_{sum} is defined as follows

$$I^i_{V_{sum}}(t) = \frac{V_{sum}(t,i)}{\mathbf{w}_i^T \mathbf{R}_e \mathbf{w}_i \chi^2_\alpha(\frac{t-t_f}{\Delta})}$$

4.5 Reconstruction of Faulty Sensors

After the isolation of faulty sensors, the next and the last step in FDD is the reconstruction of the sensors or actuators. Reconstruction of faulty sensor values is also an important step. In many industrial processes, on-line advanced control or optimization is based on the sensor values. Feeding of faulty sensor values to control loops affect the process performance, significantly degrading performance in most cases.

The basis for sensor reconstruction are the correlations among process variables. Based on the normal values of other sensors, normal readings of the faulty sensors can be estimated. The first step in sensor reconstruction is the estimation of fault magnitude based on the fault direction and the faulty data. After the occurrence of sensor fault the primary residual vector, from Equation 4.11, is

$$\mathbf{e}(t) = \mathbf{e}^*(t) + \mathbf{U}_i \mathbf{D}_i \mathbf{V}_i^T \mathbf{f}_{z,i}(t)$$

As the fault direction is known after the identification of faulty sensor, the fault magnitude is estimated by minimizing

$$J = \| \mathbf{e}^*(t) \|^2 = \| \mathbf{e}(t) - \mathbf{U}_i \mathbf{D}_i \mathbf{V}_i^T \mathbf{f}_{z,i}(t) \|^2$$

A least square solution to the above problem gives

$$\hat{\mathbf{f}}_{z,i}(t) = \left(\mathbf{U}_i \mathbf{D}_i \mathbf{V}_i^T\right)^+ \mathbf{e}(t)$$

where $()^+$ stands for the pseudo inverse.

4.6 Case Studies

This section considers the application of DSRAMS for the detection and identification of single and simultaneous multiple sensor faults in a simulated second order dynamic system. Experiment evaluation by an on-line application of DSRAMS on a pilot scale plant for the detection and identification of single sensor faults is also included.

4.6.1 Case Study One - Simulation Example

The second order dynamic system consists of 2 inputs and 5 outputs. A combination of sine wave, pulse wave and random sequence are used as noise free inputs $\tilde{\mathbf{u}}(t)$ for the generation of noise free outputs $\tilde{\mathbf{y}}(t)$. The state space model used is as follows

$$\mathbf{x}(t+1) = \begin{bmatrix} 0.67 & 0.67 \\ -0.67 & 0.67 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} -0.4326 & 0.1253 \\ -1.6656 & 0.2877 \end{bmatrix} \tilde{\mathbf{u}}(t) + \mathbf{p}(t)$$

$$\tilde{\mathbf{y}}(t) = \begin{bmatrix} 0.3273 & -0.5883 \\ 0.1746 & 2.1832 \\ -0.1867 & -0.1364 \\ 0.7258 & 0.1139 \\ 0.1324 & -0.6890 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 1.0668 & 0.2944 \\ 0.0593 & -1.3362 \\ -0.0956 & 0.7143 \\ -0.8323 & 1.6236 \\ -0.9845 & 1.2342 \end{bmatrix} \tilde{\mathbf{u}}(t)$$

where $\mathbf{p}(t) \in \Re^2$ is a normally distributed random vector having a zero mean and a variance of 0.01 for each element. The measured inputs and outputs are

$$\begin{aligned} \mathbf{u}(t) &= \tilde{\mathbf{u}}(t) + \mathbf{v}(t) \\ \mathbf{y}(t) &= \tilde{\mathbf{y}}(t) + \mathbf{o}(t) \end{aligned}$$

where $\mathbf{v}(t) \in \Re^2$ and $\mathbf{o}(t) \in \Re^5$ are normally distributed noise vectors with zero mean and the covariance matrix as $0.1^2\mathbf{I}_2$ and $0.1^2\mathbf{I}_5$ respectively. The above state space model is used for generating a data set of 1500 points. The first 800 data points are used as the training data set. The EIV-subspace algorithm is used for the identification of the residual model $\mathbf{\Gamma}_s^{\perp}\tilde{\mathbf{H}}_s \in \Re^{(m_s-n)\times(m_s+l_s)}$.

For the design of the transformation matrix, the following incidence matrix is used

	$y_1(t)$	$y_2(t)$	$y_3(t)$	$y_4(t)$	$y_5(t)$	$u_1(t)$	$u_2(t)$
$r_1(t)$	0	0	0	0	1	1	1
$r_2(t)$	1	0	0	0	0	1	1
$r_3(t)$	1	1	0	0	0	0	1
$r_4(t)$	1	1	1	0	0	0	0
$r_5(t)$	0	1	1	1	0	0	0
$r_6(t)$	0	0	1	1	1	0	0
$r_7(t)$	0	0	0	1	1	1	0

The structure of the incidence matrix is same for the time lagged values, $(t-1), \cdots, (t-s)$. In order to achieve the required transformation matrix, based on the above incidence matrix, the following condition

$$m \times (s+1) - n \geq \text{(Number of zeros in a single row of incidence matrix)} + 1 \quad (4.15)$$

must be satisfied. As the single row of the incidence matrix contains $4 \times (s+1)$ zeros and , as $n = 2$ and $m = 5$, s is chosen to be 2 for the above condition to be satisfied. The transformation matrix is then designed for the generation of structured residuals, i.e.

$$\mathbf{W} = \begin{bmatrix} \mathbf{w}_1^T \\ \vdots \\ \mathbf{w}_7^T \end{bmatrix} \in \Re^{7 \times 13}$$

The transformation matrix contains 7 rows corresponding to seven output and input variables. The sensors are numbered from 1 to 7 where sensor 1 to 5 correspond to the output variables and sensors 6 and 7 correspond to the input variables 1 and 2 respectively.

Faulty data is generated by introducing four types of faults: bias, drift, complete failure and precision degradation. The faults are simulated as follows:

$$\begin{aligned} \text{bias} &: f_i(t) = a \\ \text{drift} &: f_i(t) = b(t - t_f) \\ \text{complete failure} &: z_i(t) = c \\ \text{precision degradation} &: f_i(t) \sim \aleph(0, \sigma^2) \end{aligned}$$

for i $= 1, \cdots, 7$ and $t \geq t_f$ where t_f is the fault introduction time. The parameters a, b, c and σ are constants.

An EWMA filter with a coefficient $\gamma = 0.9$ is applied to the residuals for fault detection. After the alarm is triggered by the fault detection index, $\bar{d}(t)$, three indices, $I_{FSR}, I_{Q_{sum}}$ and $I_{V_{sum}}$, are used for the isolation of faulty sensor.

Single sensor Fault

The following table shows the fault time, fault size and the fault detection time.

Faulty sensor	1	2	6	7
Fault magnitude	$\sigma = 0.5$	$a = 0.75$	$b = .0033$	$c = -1$
Fault time(t_f)	1001	1001	1001	1001
Detection time	1002	1002	1032	1001

Figures 4.1 to 4.4 show the fault detection, isolation and reconstruction results for single sensor faults. In each of the figures, sub-plot (a) shows the fault detection index, sub-plot (c) the faulty data, sub-plot (e) the estimated fault and sub-plot (g) the reconstructed data respectively. In addition, sub-plots (b), (d) and (f) show the response of the three indices to the fault in a single sensor.

The fault is identified based on the three indices. A binary code is constructed based on the response of one index to the fault. A "0" is assigned to an element of the code if the corresponding index value is less than 1 and "1" is assigned if it is greater than 1. For instance, in Figure 4.1, the binary code corresponding to the index I_{FSR} is [0 0 1 1 0 0 0]. In addition, binary codes corresponding to $I_{Q_{sum}}$ and $I_{V_{sum}}$ are [0 0 1 0 0 0 0] and [0 1 1 1 0 0 0] respectively. In order to isolate the faulty sensor, a logic "OR" operation is performed on the three binary codes and the the resulting binary code is [0 1 1 1 0 0 0]. Since such a code matches the incidence vector for $y_1(t)$, it can be inferred that the sensor 1 is faulty. Similarly, one can use the same logic to analyze the isolation results given by Figures 4.2 to 4.4 respectively.

From Figures 4.1 to 4.4, it can be observed that the indices I_{FSR} and $I_{Q_{sum}}$ are effective at faults like bias, drift and complete failure. The index $I_{V_{sum}}$ is particularly effective at precision degradation which incurs variance change only. Therefore, to identify different types of sensor faults effectively, one needs to use all the three indices jointly.

Simultaneous multiple sensor faults

The following table shows the fault time, fault size and the fault detection time.

Faulty sensor	1 and 7	3 and 6	4 and 5	2 and 7
Fault magnitude	$b = .004$ $a = 1$	$c = 5$ $\sigma = 0.5$	$a = 2$ $\sigma = 0.5$	$b = 0.002$ $c = 3$
Fault time(t_f)	1001	1001	1001	1001
Detection time	1001	1001	1001	1002

Figures 4.5 to 4.8 show the fault detection, isolation and reconstruction results for simultaneous multiple (two) sensor faults. In each of the figures, the sub-plot (a) shows the fault detection index, sub-plots (e) and (f) the faulty data, sub-plots (g) and (h) the estimated faults and sub-plots (i) and (j) show the reconstructed signals respectively. In addition, sub-plots (b), (c) and (d) show the response of the three indices when there are faults in two sensors simultaneously.

Again, in order to isolate the faulty sensors, a logic "OR" combination of all the three binary codes, corresponding to the three indices, is used. For instance, in Figure 4.5, the binary code corresponding to the index I_{FSR} is [0 1 1 1 0 0 0]. Similarly, binary codes corresponding to $I_{Q_{sum}}$ and $I_{V_{sum}}$ are [1 1 1 1 0 0 0] and [0 0 1 1 0 0 0] respectively. The resultant binary code, which is obtained by performing an "OR" operation on the three binary codes, is [1 1 1 1 0 0 0]. From the incidence matrix, it can be inferred that the resultant binary code is a logic "OR" combination of the incidence vectors corresponding to $y_1(t)$ and $u_2(t)$. Hence the faulty sensors are identified as 1 and 7. The sub- plots for the fault estimates indicate that there is a drift in sensor 1 and a bias in sensor 7 respectively. The same logic is used for analyzing the sensor fault isolation results given by Figures 4.6 to 4.8.

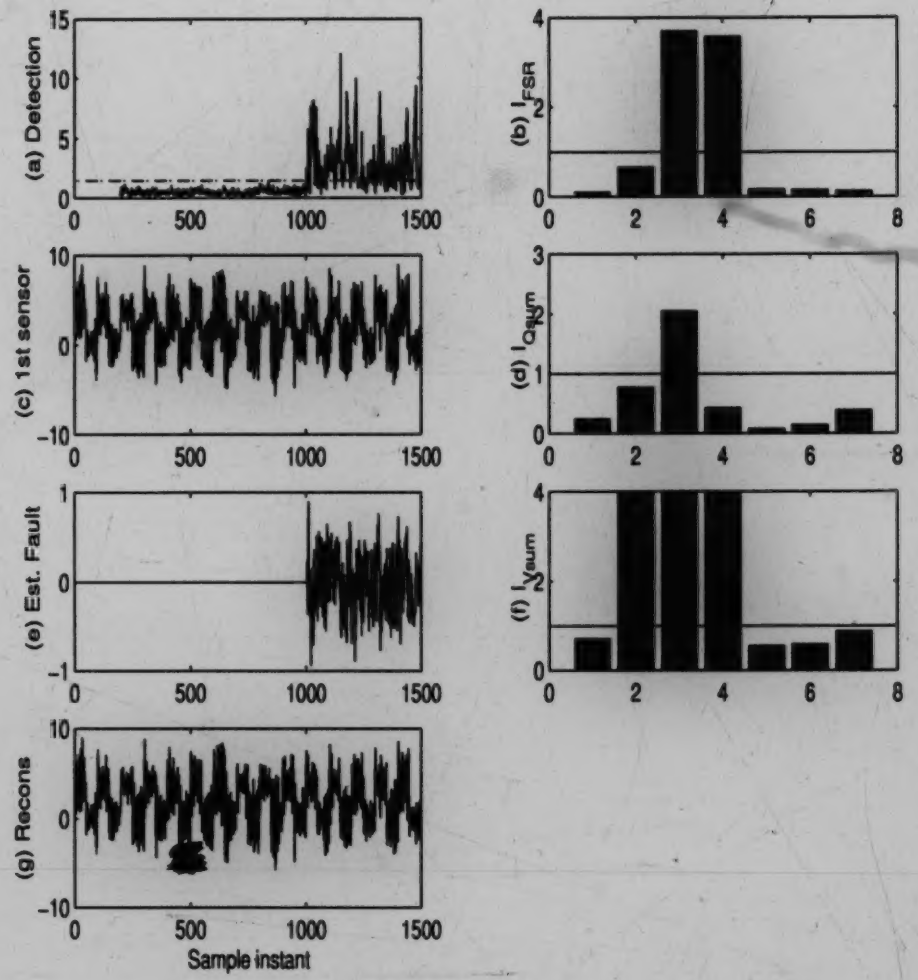


Figure 4.1: Precision degradation fault in sensor 1

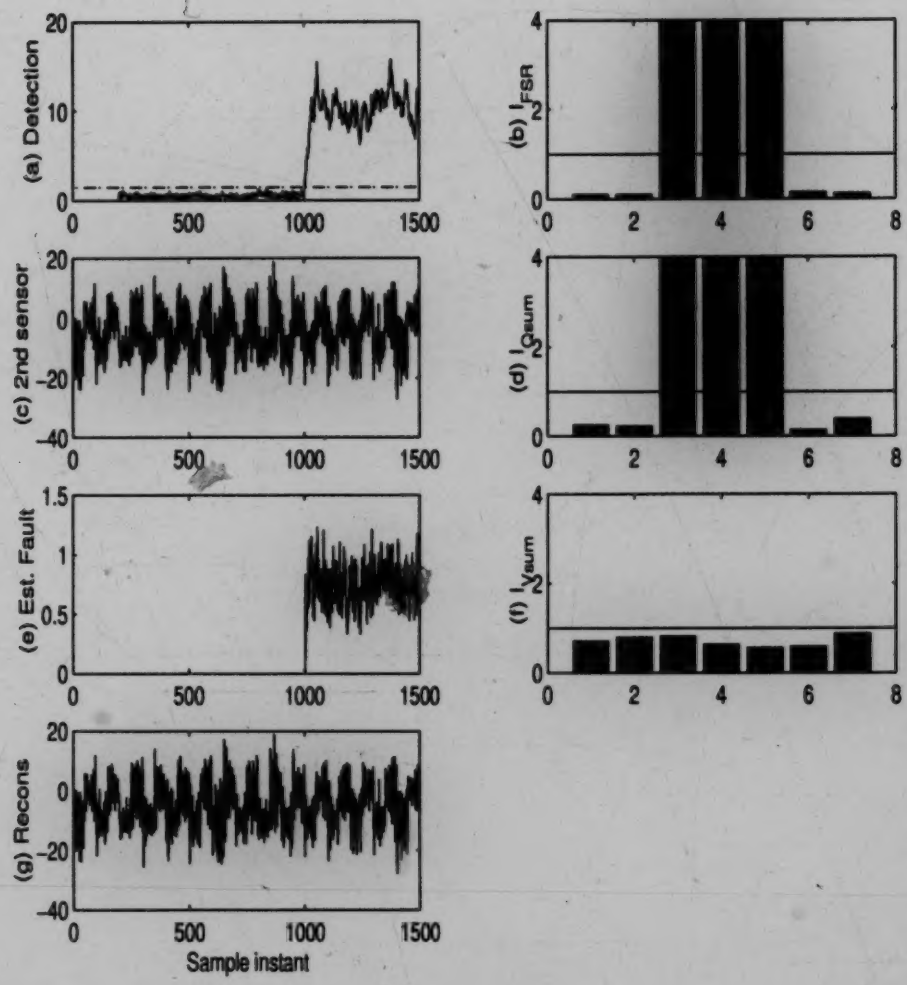


Figure 4.2: Bias fault in sensor 2

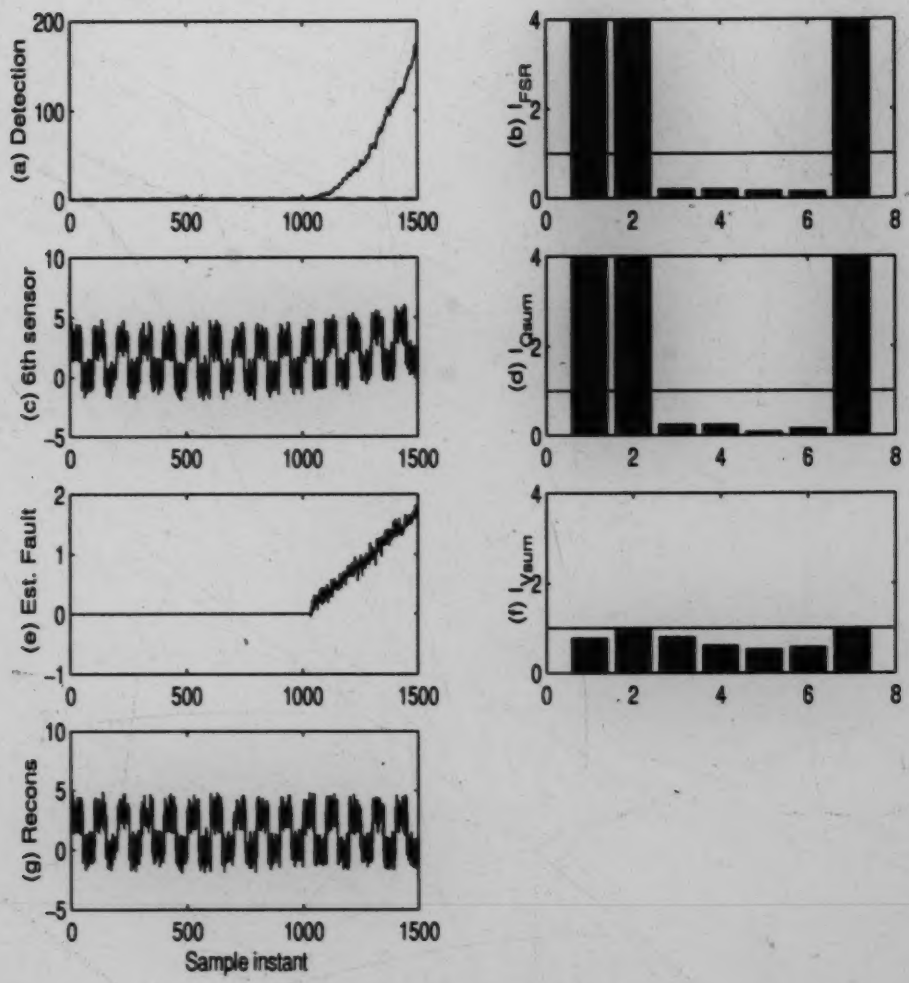


Figure 4.3: Drift fault in sensor 6

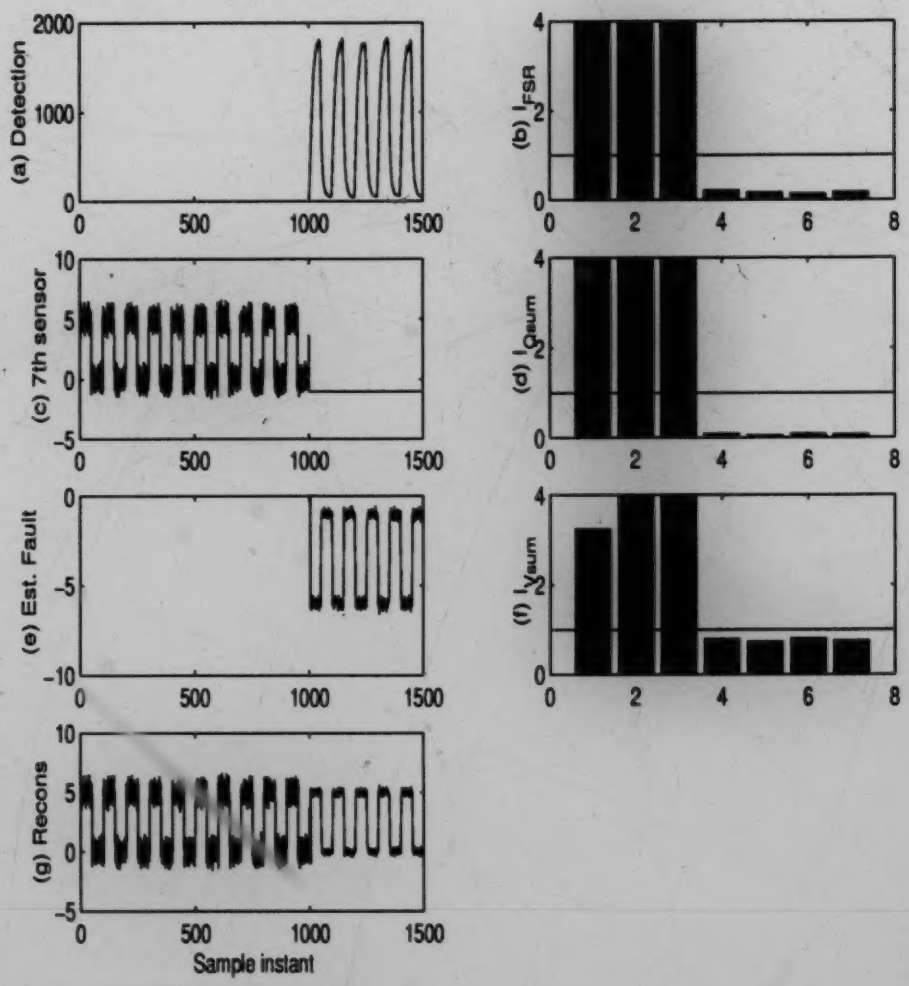


Figure 4.4: Complete failure fault in sensor 7

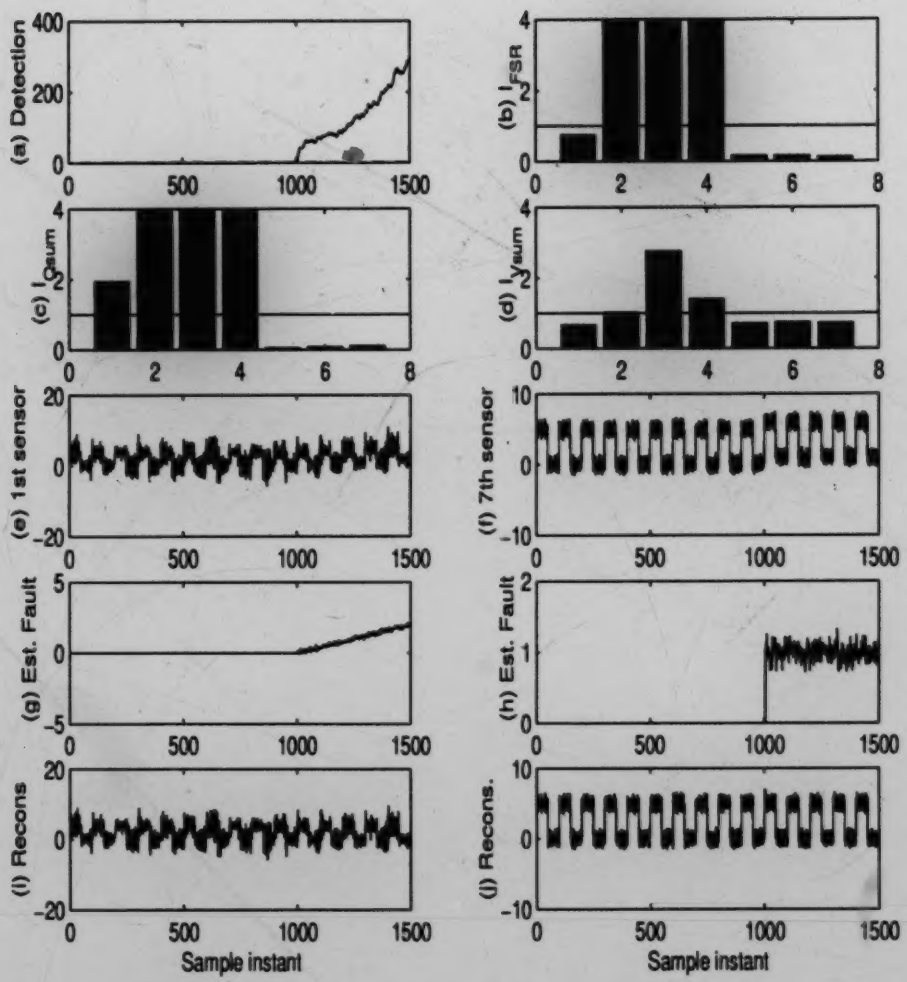


Figure 4.5: Simultaneous sensor fault detection: Drift in sensor 1 and bias in sensor 7

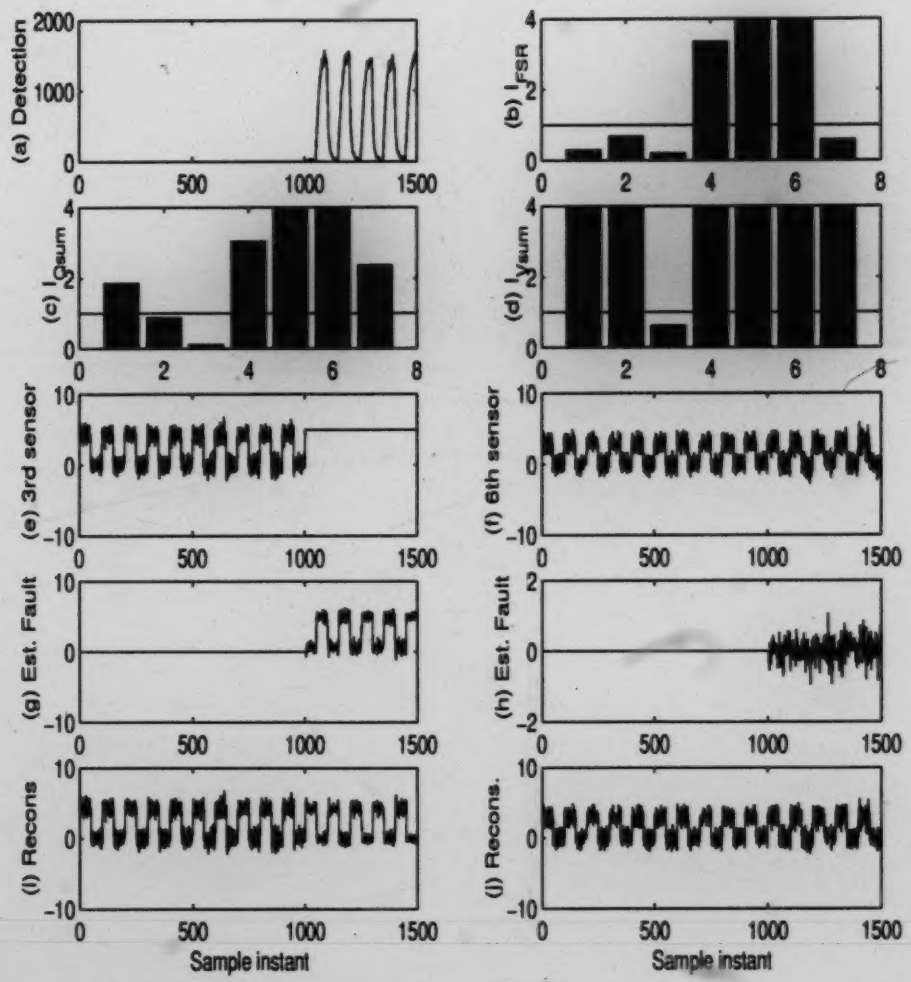


Figure 4.6: Simultaneous sensor fault detection: Complete failure in sensor 3 and precision degradation in sensor 6

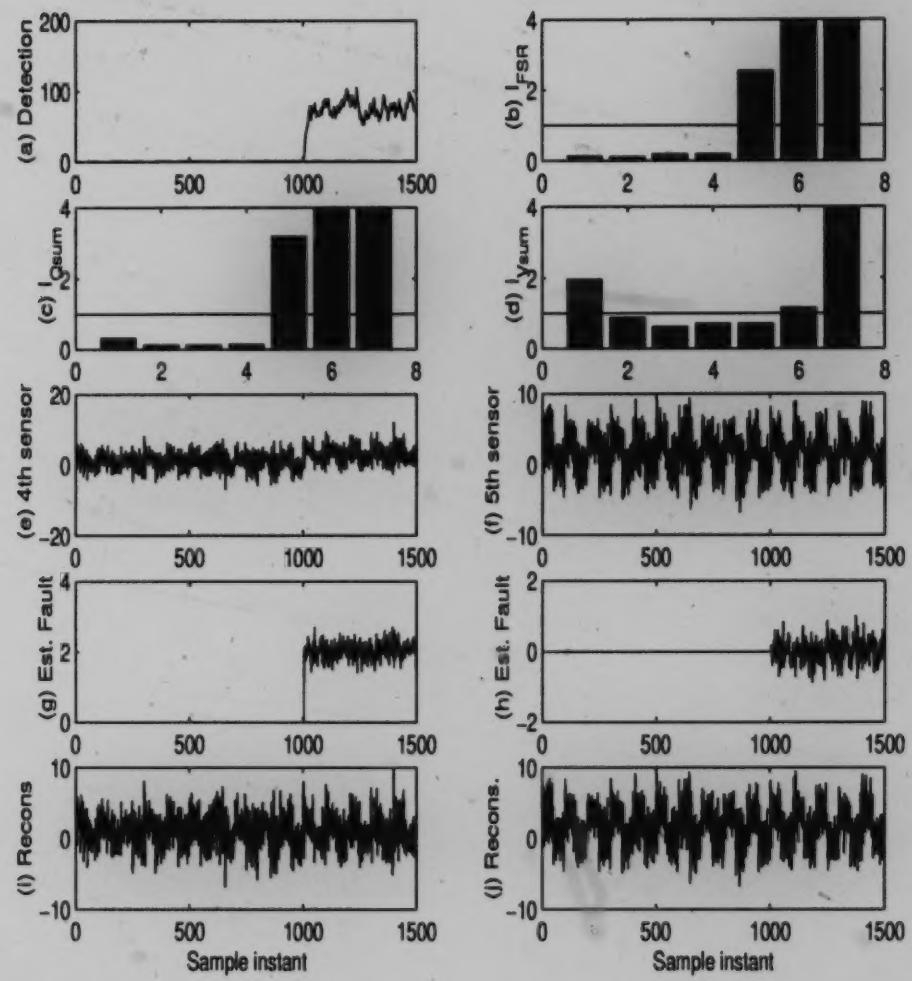


Figure 4.7: Simultaneous sensor fault detection: Bias in sensor 4 and precision degradation in sensor 5

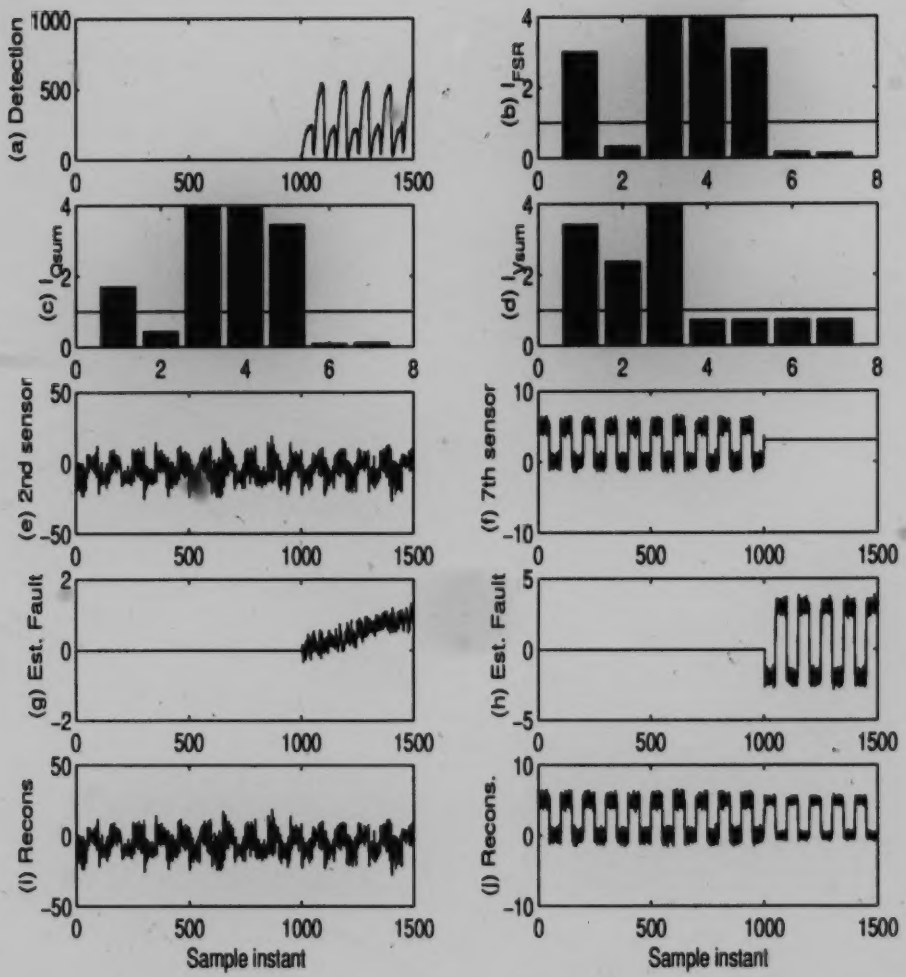


Figure 4.8: Simultaneous sensor fault detection: Drift in sensor 2 and complete failure in sensor 7

4.6.2 Case Study Two - Pilot Scale Plant

In this case study, an On-line application of DSRAMS on a pilot scale continuous stirred tank heater system is considered (refer to Figure 3.8 for schematic diagram of the process). However, in this study, DSRAMS is applied to an open loop system which consists of two output variables and two pre-determined input variables. The two output variables are the level and the temperature which are designated as sensors 1 and 2 respectively. The two input variables are the cold water valve and steam valve positions which are designated as sensors 3 and 4 respectively.

The level and the temperature are measured with a sampling period of 2 seconds. The first 350 data points are used as the training data set. Again, as in the previous case study, EIV-subspace algorithm is used for the identification of the residual model $\mathbf{\Gamma}_s^{\perp}\tilde{\mathbf{H}}_s \in \Re^{(m_s-n)\times(m_s+l_s)}$.

For the identification of single sensor faults, the following incidence matrix is used

	$y_1(t)$	$y_2(t)$	$u_1(t)$	$u_2(t)$
$r_1(t)$	0	1	1	1
$r_2(t)$	1	0	1	1
$r_3(t)$	1	1	0	1
$r_4(t)$	1	1	1	0

The structure of the incidence matrix is same for the time lagged values, $(t-1),\cdots,(t-s)$. The order of the system, n, is 3 and, in order to satisfy the condition in Equation 4.15, s is taken as 3. The following table shows the fault time, fault size and the fault detection time.

Faulty sensor	1	2	3	4
Fault magnitude	$a=1$	$\sigma=0.2$	$c=5$	$b=.01$
Fault time(t_f)	401	401	401	401
Detection time	401	404	402	420

Figures 4.9 to 4.12 show the fault detection, isolation and reconstruction results for single sensor faults. In each of the figures, sub-plot (b), (d) and (f) show the response of the three indices to the fault in a single sensor. The sub-plot (a) shows the fault detection index, sub-plot (c) shows the faulty data, sub-plot (e) is the plot of the estimated fault and sub-plot (g) shows the reconstructed data.

From Figures 4.9 to 4.12, as in the previous case study, it can be observed that the indices I_{FSR} and $I_{Q_{sum}}$ are effective in case of bias, drift and complete failure in the sensors. The index $I_{V_{sum}}$ is particularly effective when there is a precision degradation in the sensor.

Application of DSRAMS on fully dynamic experimental data

In the above case study on the experimental data, DSRAMS was applied on the pilot scale plant under steady state conditions. In this section, an off line evaluation of DSRAMS on the same pilot scale plant is carried out but under fully dynamic conditions.

A total of 1000 data points are collected with a sampling period of 2 seconds under fully dynamic conditions. The first 650 data points are used as the training data set. The following table shows the fault time, fault size and the fault detection time.

Faulty sensor	1	2	3	4
Fault magnitude	$a = 2$	$c = 5$	$a = 5$	$\sigma = 4$
Fault time(t_f)	701	701	701	701
Detection time	701	701	702	704

Figures 4.13 to 4.16 show the fault detection, isolation and reconstruction results for single sensor faults. The same logic, as in the above case study, is used to analyze the sensor fault isolation results. DSRAMS is able to isolate the faulty sensor when the fault magnitude is large but is ineffective for small fault magnitude. For instance, DSRAMS is effective in isolating the bias of magnitude 5 in the 3rd sensor (Figure 4.15) but is unable to isolate the faulty sensor when there is a bias of magnitude 2 in sensor 1 (Figure 4.13). Also the reconstruction of faulty sensors is poor as can be seen in Figures 4.14 and 4.16.

4.7 Summary

The EIV subspace algorithm has been used for the identification of the residual model from the noisy input and output data. Filtered residuals are used for the detection of faults. DSRAMS has been used for the generation of structured residuals. For the isolation of faults, three indices $I_{FSR}, I_{Q_{sum}}$ and $I_{V_{sum}}$, have been jointly used. Fault detection and isolation scheme has been successfully applied to a 2×5 second order dynamic system where four different types of faults are simulated. Also an on-line application of DSRAMS on a pilot scale plant has been successfully carried out where single sensor faults are detected and identified. Also DSRAMS when applied to fully dynamic experimental data gave mixed results. A further investigation needs to be carried out to improve the effectiveness of DSRAMS for the isolation of faults in fully dynamic real systems.

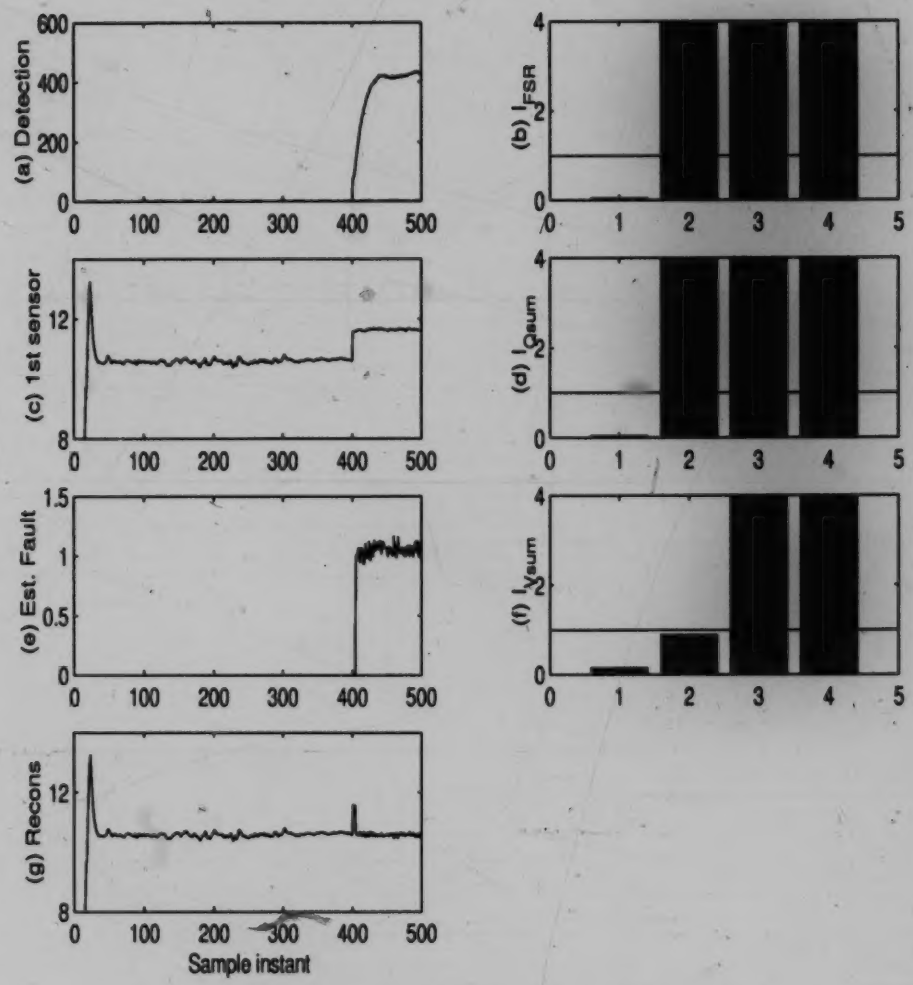


Figure 4.9: Bias in sensor 1 (level)

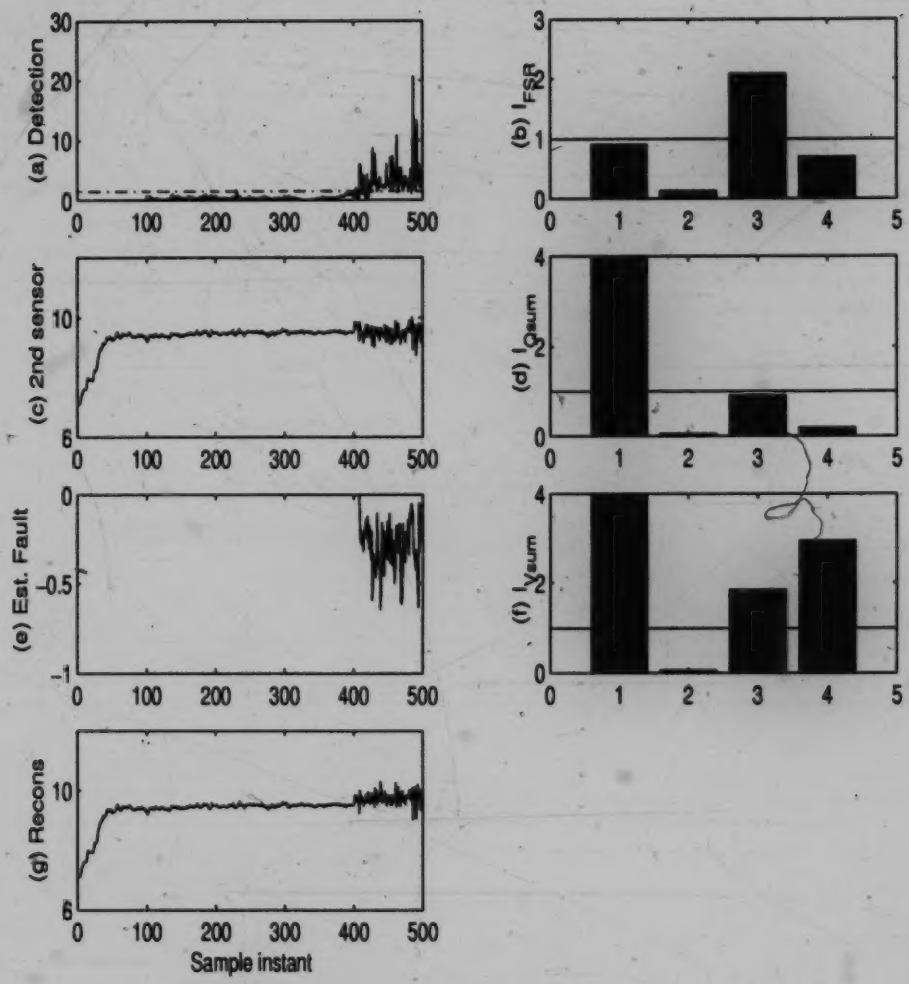


Figure 4.10: Precision degradation in sensor 2 (temperature)

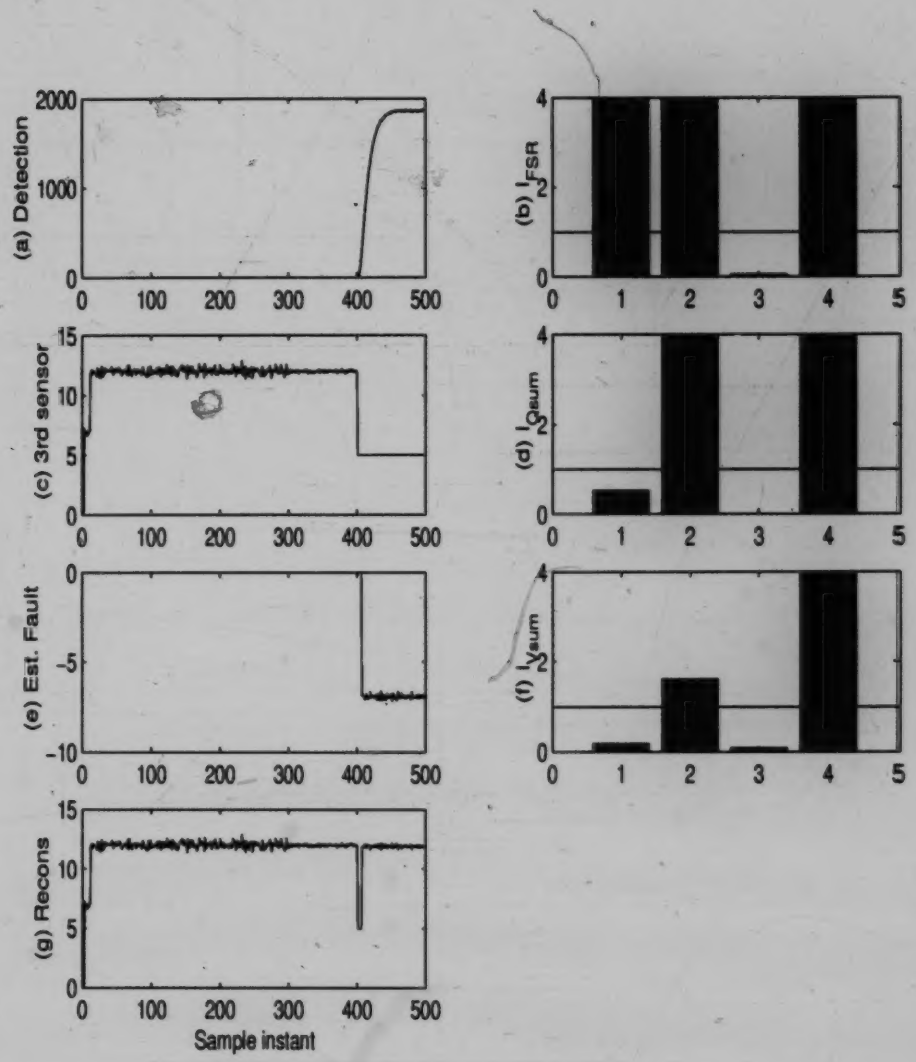


Figure 4.11: Complete failure in sensor 3 (Steam valve)

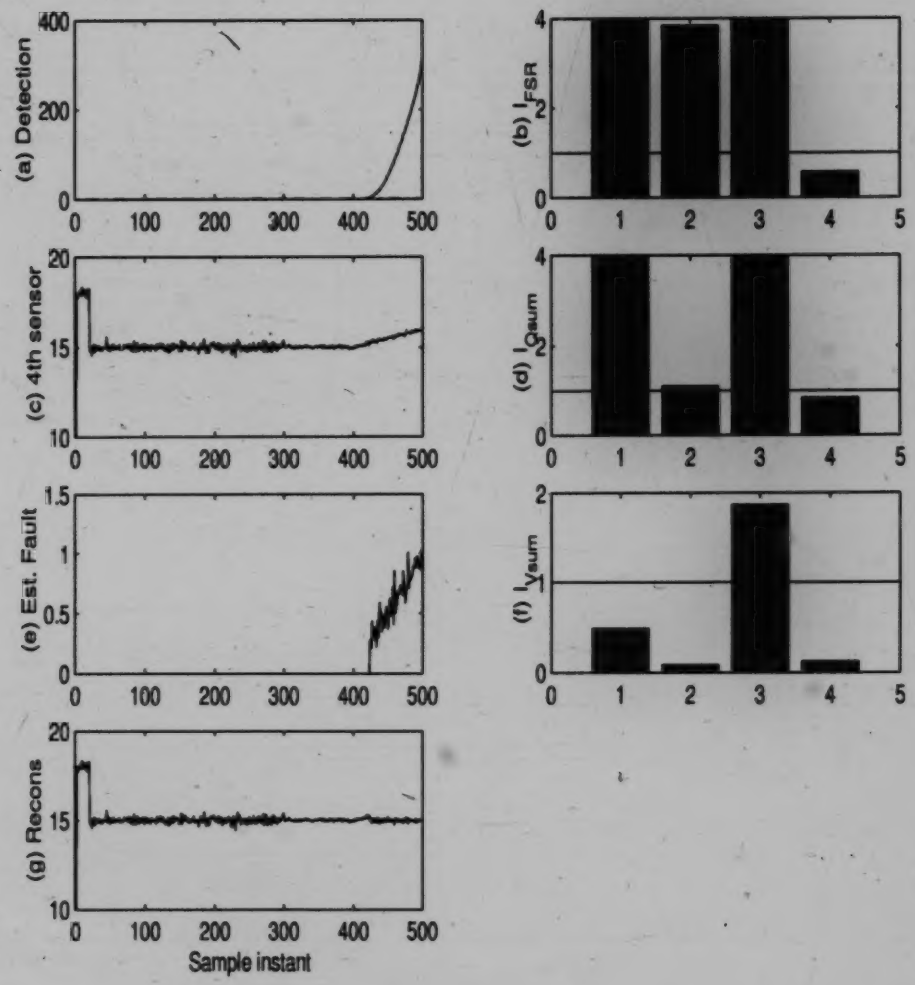


Figure 4.12: Drift in sensor 4 (Cold water valve)

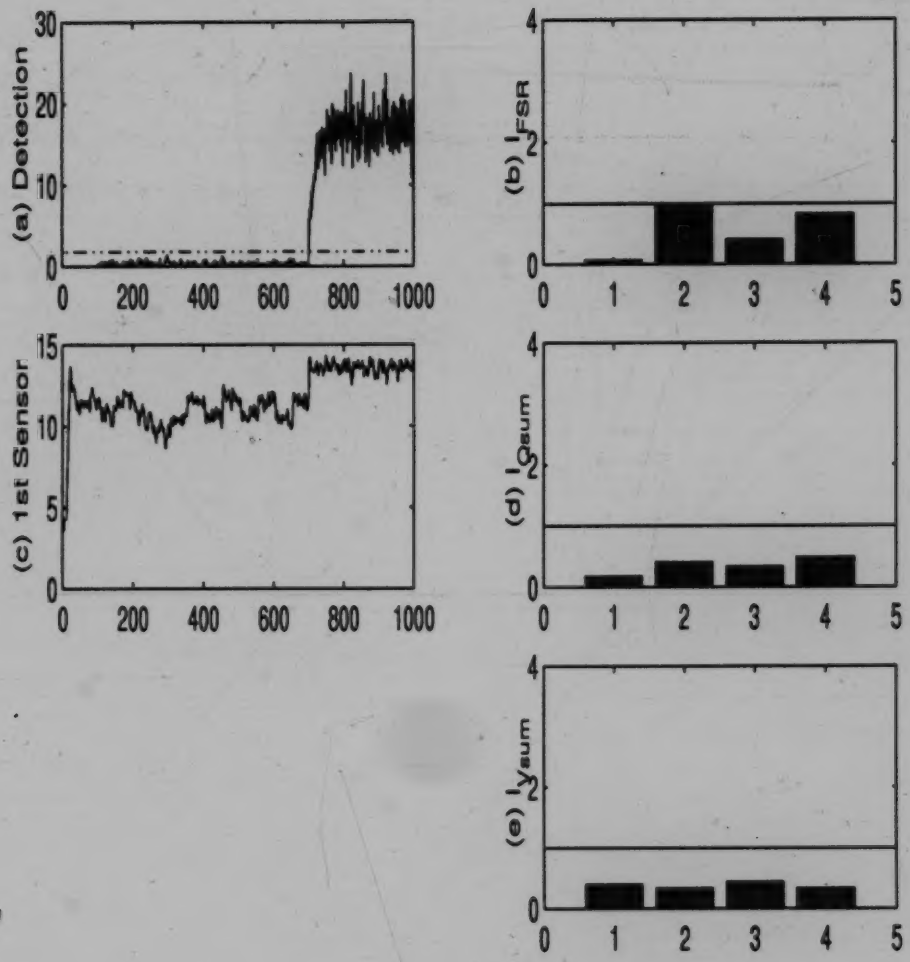


Figure 4.13: Bias in sensor 1 (level)

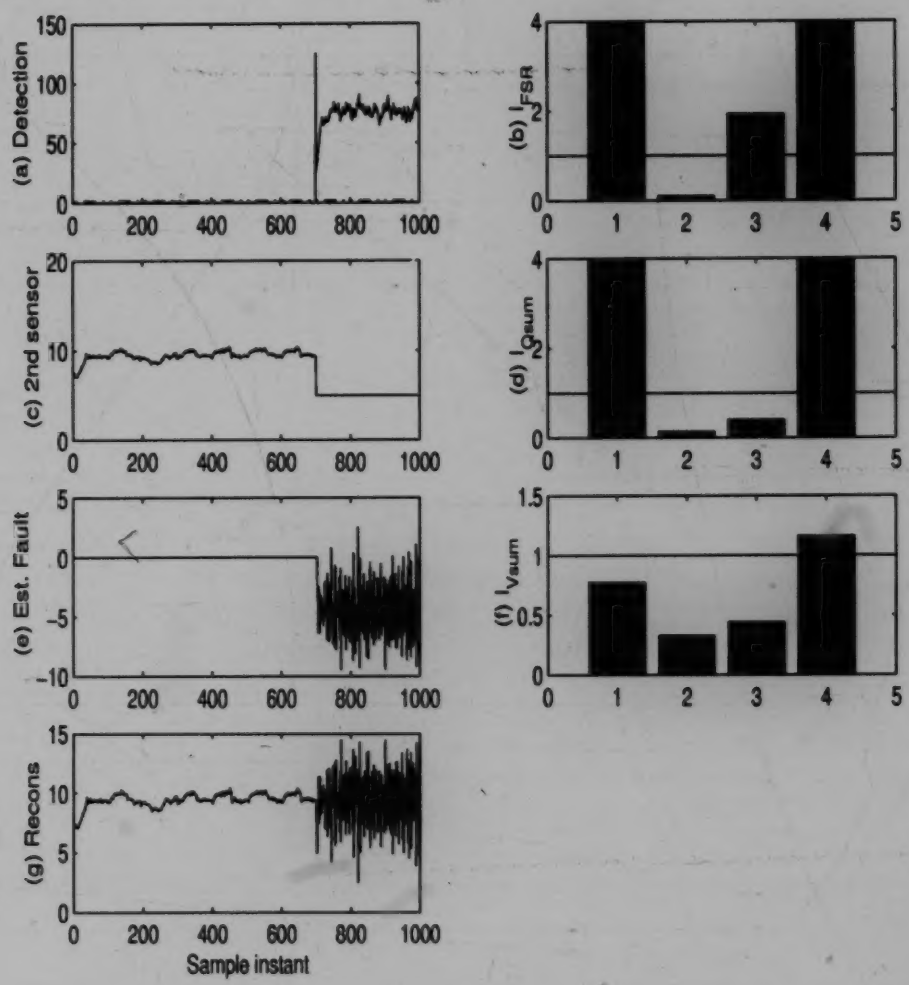


Figure 4.14: Complete failure in sensor 2 (temperature)

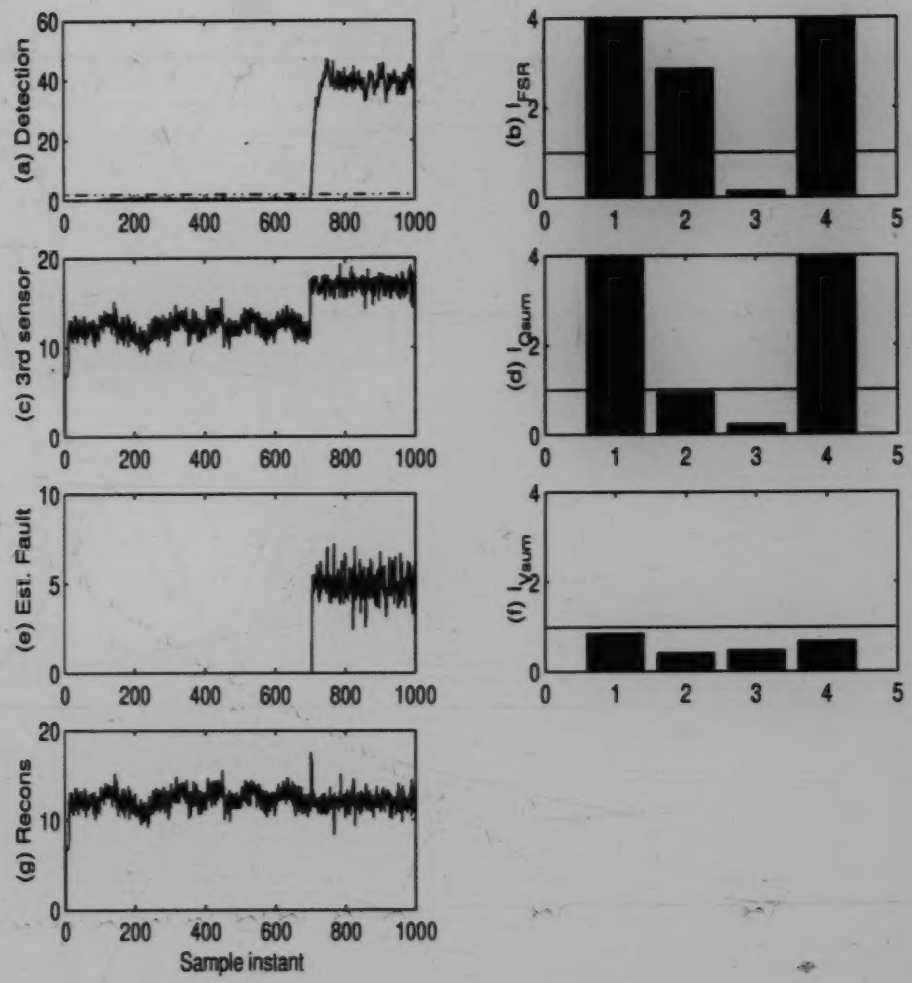


Figure 4.15: Bias in sensor 3 (Steam valve)

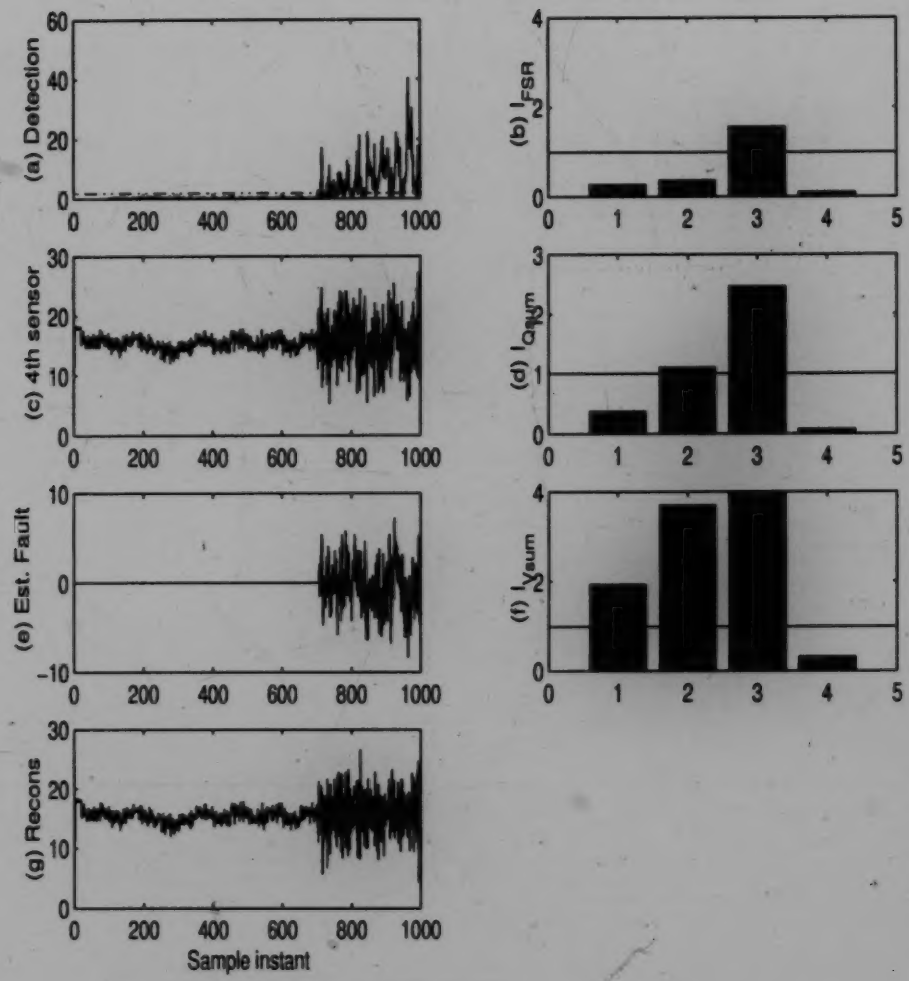


Figure 4.16: Precision degradation in sensor 4 (Cold water valve)

Chapter 5

Summary and Conclusions

Process monitoring and fault detection and diagnosis is extremely important for plant safety and for maintaining good product quality. Many chemical processes vary slowly with time which makes it imperative to have an effective technique for adaptive process monitoring to reduce false alarm rates. Equally important is the proper functioning of sensors which provide a crucial link to the process status. Sensor failures can cause process disturbances, loss of control or catastrophic accidents. Hence, a reliable sensor fault detection and isolation mechanism is of prime importance to the control systems that rely solely on the information provided by the sensors for decision making.

This thesis proposes a novel method for adaptive process monitoring of multivariate dynamic systems via the use of multi channel lattice filter. The most important advantage of using a lattice filter is that it is recursive not only in time but also in order. The multivariate dynamic process is represented by EIV state space model where input and output measurement noise as well as process noise are taken into account. By showing the relationship between EIV state space representation and multi channel lattice filters, the latter is used to generate a residual vector for adaptive process monitoring. The computed residual vector is used for constructing the Hotelling T^2 statistic which is used as a monitoring index. The effectiveness of the proposed scheme has been demonstrated on a simulated process and a real pilot scale plant. The proposed scheme not only tracks the slowly time varying parameters of the process effectively but is also sensitive to faults.

For the detection of faulty sensors, the EIV subspace algorithm is used for the identification of the residual model. The sensor fault isolation is carried out using the *dynamic structured residual approach with maximized sensitivity* (DSRAMS) as proposed by Qin and Li (2000). The DSRAMS approach was proposed for the isolation of single sensor faults. Their approach has been extended to the isolation of simultaneous multiple (two) sensor faults. DSRAMS has been applied successfully on a simulated process for the isolation of four different types of faults. DSRAMS has also been applied on-line on a real pilot scale plant and encouraging results have been obtained.

In summary, the main contribution of this thesis are:

- A novel method for adaptive process monitoring via the use of multi-channel lattice filter has been proposed. The effectiveness of the proposed scheme has been demonstrated on a simulated process and on a pilot scale plant.

- Evaluation of DSRAMS for the isolation of single and multiple sensor faults has been carried out on a simulated process. DSRAMS is applied on-line on a pilot scale plant for the isolation of single sensor faults under steady conditions.

5.1 Directions for Future Work

- In the proposed scheme, residuals generated via multi channel lattice filter are updated at each data sample. In practical situation particularly for slowly time varying chemical process, the process model need not be updated at each data sample. Instead it is suggested that the model be updated based on a non-overlapping data blocks of user specified window lengths.

- A fixed forgetting factor, $\lambda = 0.99$, has been used for adaptive process monitoring via multi channel lattice filter. The effect of time varying forgetting factor on the adaptive process monitoring needs to be investigated.

- A further investigation is needed to improve the effectiveness of DSRAMS for the isolation of sensor faults in fully dynamic real systems.

- The results for isolation of single and multiple sensor faults have been presented for open loop systems. In open loop systems a fault in one sensor does not propagate through out the system i.e., the sensor fault is localized. The real challenge lies in isolation of sensor faults under closed loop conditions where a fault in one sensor propagates throughout the system.

Bibliography

Albuquerque, J. S. and L. T. Biegler (1996). Data reconciliation and gross-error detection for dynamic systems.. *AIChE J.* **42**, 2841–2854.

Bakshi, B. R. (1998). Multiscale PCA with application to multivariate statistical process monitoring. *AIChE J.* **44**(7), 1596–1610.

Basseville, M. (1997). Information criteria for residual generation and fault detection and isolation.. *Automatica* **33**(5), 783–803.

Basseville, M. (1998). On-board component fault detection and isolation using the statistical local approach. *Automatica* **34**(11), 1391–1415.

Basseville, M. and I. V. Nikiforov (1993). *Detection of Abrupt Changes-Theory and Applications*. Prentice-Hall, Englewood Cliffs, NJ.

Basseville, M., M. Abdelghani and A. Benveniste (2000). Subspace-based fault detection algorithms for vibration monitoring. *Automatica* **36**(1), 101–109.

Beard, R. V. (1971). *Failure Accommodation in Linear Systems Through Self-Reorganization*. Dept. MVT-71-1, Man Vehicle Laboratory, Cambridge, MA.

Benveniste, A., M. Basseville and G. Moustakides (1987). The asymptotic local approach to change detection and model validation. *IEEE Trans. Auto. Cont.* **32**(7), 583–592.

Box, G. E. P. and G. M. Jenkins (1976). *Time Series Analysis: Forecasting and Control*. Holden-Day, San Francisco, California.

Burnham, A. J., J. F. MacGregor and R. Viveros (1999). Latent variable multivariate regression modeling. *Chemometrics and Intelligent Laboratory Systems* **48**, 167–180.

Chen, G. and T. J. McAvoy (1998). Predictive on-line monitoring of continuous processes. *J. Process Control* **8**(5-6), 409–420.

Chen, H. F. and L. Guo (1987). Consistent estimation of the order of stochastic control systems. *IEEE Trans. Auto. Cont.* **32**(6), 531–535.

Chou, C. T. and M. Verhaegen (1997). Subspace algorithms for the identification of multivariable dynamic errors-in-variables models. *Automatica* **33**(10), 1857–1869.

Chow, E. Y. and A. S. Willsky (1984). Analytical redundancy and the design of robust failure detection systems. *IEEE Trans. Auto. Cont.* **29**, 603–614.

Clark, R. N. (1979). The dedicated observer approach to instrument fault detection. *Proceedings of the 15th IEEE-CDC, Fort Lauderdale, FL*. pp. 237–241.

Clark, R. N., D. C. Fosth and V. M. Walton (1975). Detection instrument malfunctions in control systems. *IEEE Trans. Aerospace Electron. Syst.* **AES-11**, 465–473.

Crowe, C. M. (1996). Data reconciliation-progress and challenges. *J. Process Control* **6**, 89–98.

Deckert, J. C., M. N. Desai, J. J. Deyst and A. S. Willsky (1977). F-8 DFBW sensor failure identification using analytical redundancy. *IEEE Trans. Auto. Cont.* **22**, 796–803.

Ding, X., L. Guo and T. Jeinsch (1999). A characterization of parity space and its application to robust fault detection. *IEEE Trans. Auto. Cont.* **44**(2), 337–343.

Dunia, R. and S. J. Qin (1998). A unified geometric approach to process and sensor fault identification and reconstruction: The unidimensional fault case. *Computers and Chemical Engineering* **22**(7-8), 927–943.

Dunia, R., S. J. Qin, T. F. Edgar and T. J. McAvoy (1996). Identification of faulty sensors using principal component analysis. *AIChE J.* **42**, 2797–2812.

Frank, P. M. (1990). Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy - a survey and some new results. *Automatica* **26**(3), 459–474.

Frank, P. M. and J. Wunnenberg (1989). *Robust fault diagnosis using unknown input observer schemes, in fault diagnosis in dynamic systems.* Prentice-Hall, Englewood Cliffs, NJ.

Friedlander, B. (1982). Lattice methods for adaptive processing. *Proc. IEEE* **70**(8), 829–867.

Friedlander, B. (1983). Lattice implementation of some recursive parameter-estimation algorithms. *Int. J. Control* **37**(4), 661–684.

Gertler, J. (1988). Survey of model-based failure detection and isolation in complex plants. *IEEE Cont. Sys. Mag.* **12**, 3–11.

Gertler, J. (1991). Analytical redundancy method in fault detection and isolation-survey and synthesis. *Proceedings of the IFAC SAFEPROCESS Symposium* pp. 10–13.

Gertler, J. and D. Singer (1985). Augmented models for statistical fault isolation in complex dynamic systems.. *Proceedings of the American Control Conference* pp. 317–322.

Gertler, J. and D. Singer (1990). A new structural framework for parity equation based failure detection and isolation. *Automatica* **26**, 381–388.

Gertler, J. and M. M. Kunwer (1995). Optimal residual decoupling for robust fault diagnosis. *Int. J. Control* **61**, 395–421.

Gertler, J., W. Li, Y. Huang and T. McAvoy (1999). Isolation-enhanced principal component analysis. *AIChE J.* **45**(2), 323–334.

Gustafsson, T., M. Lovero and M. Verhaegen (1998). A novel algorithm for recursive instrumental variable based subspace identification. *Proc. of 34th IEEE-CDC.*

Honig, M. L. and D. G. Messerschmitt (1981). Convergence properties of an adaptive digital lattice filters. *IEEE Trans. Acoust. Speech, Signal Processing* **29**(3), 642–653.

Honig, M. L. and D. G. Messerschmitt (1984). *Adaptive Filters - Structures, Algorithms and Applications.* Kluwer Academic Publishers.

Isermann, R. (1984). Process fault detection based on modeling and estimation methods - a survey. *Automatica* **20**, 387–404.

Itakura, F. and S. Saito (1971). Digital filtering techniques for speech analysis and synthesis. *Proc. 7th Int. Conf. Acoust., Budapest, Hungary* pp. 261–264.

Jabbari, F. and J. S. Gibson (1988). Vector-channel lattice filters and identification of flexible structures. *IEEE Trans. Auto. Cont.*

Johnson, R. A. and D. W. Wichern (1998). *Applied Multivariate Statistical Analysis.* Prentice-Hall, Inc.

Jones, H. L. (1973). Failure detection in linear systems. PhD thesis. MIT, Cambridge, MA.

Karjala, T. W. and D. M. Himmelblau (1996). Dynamic rectification of data via recurrent neural nets and the extended kalman filter. *AIChE. J.* **42**, 2225.

Kasper, M. H. and W. H. Ray (1992). Chemometric Methods for Process Monitoring and High-Performance Controller Design. *AIChE. J.* **38**(10), 1593–1608.

Kourti, T. and J. F. MacGregor (1995). Process analysis, monitoring and diagnosis using multivariate projection methods. *Chemometrics and Intelligent Laboratory Systems* **28**, 3–21.

Kramer, M. (1991). Nonlinear principal component analysis using auto-associative neural networks. *AIChE. J.* **37**, 233.

Kresta, J. V. (1992). Applications of Partial Least Squares Regression. PhD thesis. McMaster University.

Kresta, J. V., J. F. MacGregor and T. E. Marlin (1991). Multivariate statistical monitoring of processes. *Can. J. Chem. Eng.* **69**(1), 35–47.

Ku, W., R. H. Storer and C. Georgakis (1995). Disturbance detection and isolation by dynamic principal component analysis. *Chemometrics and Intelligent Laboratory Systems* **30**, 179–196.

Kummert, A., S. Hartwig and A. Hypki (1992). A new adaptive lattice filter for the ARMA modeling of vector signals. *Signal Processing.*

Lakshminarayanan, S. (1997). Process Characterization and Control using Multivariate Statistical Techniques. PhD thesis. University of Alberta, Edmonton.

Lakshminarayanan, S., S. L. Shah and K. Nandakumar (1995). Identification of Hammerstein Models using Multivariate Statistical Tools. *Chemical Engineering Science* **50**(22), 3599–3613.

Larimore, W. E. (1983). System identification, reduced-order filtering and modeling via canonical variate analysis. *Proceedings of the 1983 American Control Conference* pp. 445–451.

Larimore, W. E. (June 1997). Process monitoring using canonical variate analysis and principal component analysis. *Proceedings of 1997 IFAC ADCHEM.*

Lee, D. T. L., M. Morf and B. Friedlander (1981). Recursive least squares ladder estimation algorithms. *IEEE Trans. Acoust. Speech, Signal Processing* **29**(3), 627–641.

Lee, D. T. L., M. Morf and B. Friedlander (1982). Recursive ladder estimation algorithms for ARMA modeling. *IEEE Trans. Auto. Cont.* **27**(8), 753–764.

Lev-Ari, H., T. Kailath and J. Cioffi (1984). Least-squares adaptive lattice and transversal filters: A unified geometric theory.. *IEEE Trans. Inform. Theo.* **30**(2), 222–236.

Liebman, M. J., T. F. Edgar and L. S. Lasdon (1992). Efficient data reconciliation and estimation for dynamic processes using nonlinear programming techniques. *Computers and Chemical Engineering* **16**(11), 963–986.

Ljung, L. (1987). *System Identification: Theory for the User.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

Luo, R., M . Misra and D. M. Himmelblau (1999). Sensor fault detection via multiscale analysis and dynamic PCA. *Ind. Eng. Chem. Res.* **38**, 1489–1495.

MacGregor, J. F. (1989). Multivariate statistical methods for monitoring large data sets from chemical processes. *AIChE Meeting, San Francisco.*

Mah, R. S. H., G. M. Stanley and D. Downing (1976). Reconciliation and rectification of process flow and inventory data. *Ind. Eng. Chem. Proc. Des. Dev.* **15**, 175.

Mehra, R. K. and I. Peshon (1971). An innovations approach to fault detection and diagnosis in dynamic systems. *Automatica* **7**, 637–640.

Miller, P., R. E. Swanson and C. F. Heckler (1993). Contribution plots: the missing link in multivariate quality control. *In Fall Conf. Of the ASQC and ASA* pp. Milwaukee, WI.

Nomikos, P. and J. F. MacGregor (1995). Multivariate SPC charts for monitoring batch processes. *Technometrics* **37**(1), 41–59.

Piovoso, M., K. Kosanovich and R. Pearson (1992). Monitoring process performance in real time. *Proceedings of ACC, Chicago* pp. 2359–2363.

Qin, S. J. and W. Li (1999). Detection, Identification, and Reconstruction of Faulty Sensors with Maximized Sensitivity. *AIChE J* **45**(9), 1963–1976.

Qin, S. J. and W. Li (2000). Detection and identification of faulty sensors in dynamic processes with maximized sensitivity. *submitted to AIChE J.*

Reddy, V. N. and M. L. Mavrovouniotis (1998). An Input-Training Neural Network approach for gross error detection and sensor replacement.. *Chem. Eng. Res. Des.* **76**(A4), 478–489.

Romagnoli, J. A. and G. Stephanopoulos (1981). Rectification of process measurement data in the presence of gross errors. *Chem. Eng. Sci.* **36**, 1849.

Schaper, C. D., W. E. Larimore, D. E. Seborg and D. A. Mellichamp (1994). Identification of Chemical Processes using Canonical Variate Analysis. *Computers and Chemical Engineering* **18**(1), 55–69.

Shao, R., F. Jia, E. B. Martin and A. J. Morris (1999). Wavelets and non-linear principal component analysis for process monitoring. *Control Engineering Practice* **7**(7), 865–879.

Stanley, G. M. and R. S. H. Mah (1977). Estimation of flows and temperature in process networks. *AIChE. J.* **23**, 642.

Stanley, G. M. and R. S. H. Mah (1981). Observability and Redundancy in Process Data Estimation. *Chem. Eng. Sci.* **36**, 259.

Tong, H. and C. M. Crowe (1995). Detection of gross errors in Data Reconciliation by Principal Component Analysis. *AIChE J.* **41**, 1712.

W. Li, H. H. Yue, S. V. Cervantes and S. J. Qin (2000). Recursive PCA for adaptive process monitoring. *J. Process Control* **10**, 471–486.

Wachs, A. and D. R. Lewin (1999). Improved PCA methods for process disturbance and failure identification. *AIChE J.* **45**(8), 1688–1700.

Wang, Y., D.E. Seborg and W. E. Larimore (1997). Process monitoring using Canonical Correlation Analysis and Principal Component Analysis. *Proceedings of 1997 IFAC ADCHEM.*

Wise, B. M. (1991). Adapting Multivariate Analysis for Monitoring and Modeling Dynamic Systems. PhD thesis. University of Washington.

Wise, B. M., N. B. Gallagher and J. F. MacGregor (1996). The process chemometrics approach to process monitoring and fault detection. *J. Process Control* **6**(6), 329–348.

Wold, S. (1994). Exponentially weighted moving principal component analysis and projection to latent structures. *Chemometrics and Intelligent Laboratory Systems* **23**, 149–161.

Wold S., P. Geladi, K. Esbensen and J. Ohman (1987). Multi-way principal components and PLS analysis. *J. Chemometrics* **1**, 41–56.

Xia, L. and J. B. Moore (1989). Recursive identification of overparameterized systems.. *IEEE Trans. Auto. Cont.* **34**, 327–331.

Ying, C. M. and B. Joseph (2000). Sensor fault detection using noise analysis. *Ind. Eng. Chem. Res.* **39**, 396–407.

Yoon, S. and J. F. MacGregor (1998). Sensor fault diagnosis for dynamic systems using multivariate statistical methods. *In Proceedings of the IFAC Workshop on On-line Fault Detection and Supervision in the Chemical Process Industries.*

Appendix A

A.1 Derivation of Equation 3.37

Based on the definition of $\mathbf{f}^i_{s,\eta}(k)$ given by Equation 3.18 and the non-symmetric projection formula given by Equation 3.24, Equation 3.37 is derived as follows

$$
\begin{aligned}
\mathbf{f}^i_{s,\eta}(k) &= \mathbf{z}^i_{kM-s} - \mathbf{z}^i_{kM-s}|_{\mathbf{h}^i_{s,\eta-1}(k) \oplus \mathrm{Span}\{\mathbf{b}^i_{s+1,\eta-1}(k)\}} \\
&= \mathbf{z}^i_{kM-s} - \left[\mathbf{z}^i_{kM-s-1:kM-s-\eta+1} \;\; \mathbf{b}^i_{s+1,\eta-1}(k)\right] \bullet \\
&\quad \left(\begin{bmatrix} \left(\mathbf{z}^i_{(k-\mu)M-s-1:(k-\mu)M-s-\eta+1}\right)^T \\ \left(\mathbf{b}^i_{s+1,\eta-1}(k-\mu)\right)^T \end{bmatrix} \mathbf{\Lambda} \left[\mathbf{z}^i_{kM-s-1:kM-s-\eta+1} \;\; \mathbf{b}^i_{s+1,\eta-1}(k)\right] \right)^{-1} \bullet \\
&\quad \begin{bmatrix} \left(\mathbf{z}^i_{(k-\mu)M-s-1:(k-\mu)M-s-\eta+1}\right)^T \mathbf{\Lambda} \mathbf{z}^i_{kM-s} \\ \left(\mathbf{b}^i_{s+1,\eta-1}(k-\mu)\right)^T \mathbf{\Lambda} \mathbf{z}^i_{kM-s} \end{bmatrix} \\
&= \mathbf{z}^i_{kM-s} - \left[\mathbf{z}^i_{kM-s-1:kM-s-\eta+1} \;\; \mathbf{b}^i_{s+1,\eta-1}(k)\right] \bullet \\
&\quad \begin{bmatrix} \left(\mathbf{z}^i_{(k-\mu)M-s-1:(k-\mu)M-s-\eta+1}\right)^T \mathbf{\Lambda} \mathbf{z}^i_{kM-s-1:kM-s-\eta+1} & 0 \\ \left(\mathbf{b}^i_{s+1,\eta-1}(k-\mu)\right)^T \mathbf{\Lambda} \mathbf{z}^i_{kM-s-1:kM-s-\eta+1} & \beta^i_{s+1,\eta-1}(k) \end{bmatrix}^{-1} \bullet \\
&\quad \begin{bmatrix} \left(\mathbf{z}^i_{(k-\mu)M-s-1:(k-\mu)M-s-\eta+1}\right)^T \mathbf{\Lambda} \mathbf{z}^i_{kM-s} \\ \left(\mathbf{b}^i_{s+1,\eta-1}(k-\mu)\right)^T \mathbf{\Lambda} \mathbf{z}^i_{kM-s} \end{bmatrix} \\
&= \mathbf{z}^i_{kM-s} - \mathbf{z}^i_{kM-s-1:kM-s-\eta+1} \left(\left(\mathbf{z}^i_{(k-\mu)M-s-1:(k-\mu)M-s-\eta+1}\right)^T \mathbf{\Lambda} \mathbf{z}^i_{kM-s-1:kM-s-\eta+1} \right)^{-1} \bullet \\
&\quad \left(\mathbf{z}^i_{(k-\mu)M-s-1:(k-\mu)M-s-\eta+1}\right)^T \mathbf{\Lambda} \mathbf{z}^i_{kM-s} \\
&\quad + \frac{\mathbf{b}^i_{s+1,\eta-1}(k)}{\beta^i_{s+1,\eta-1}(k)} \left(\mathbf{b}^i_{s+1}(k-\mu)\right)^T \mathbf{\Lambda} \mathbf{z}^i_{kM-s-1:kM-s-\eta+1}(k) \bullet \\
&\quad \left(\left(\mathbf{z}^i_{(k-\mu)M-s-1:(k-\mu)M-s-\eta+1}\right)^T \mathbf{\Lambda} \mathbf{z}^i_{kM-s-1:kM-s-\eta+1}(k) \right)^{-1} \bullet \\
&\quad \left(\mathbf{z}^i_{(k-\mu)M-s-1:(k-\mu)M-s-\eta+1}\right)^T \mathbf{\Lambda} \mathbf{z}^i_{kM-s} - \frac{\left(\mathbf{b}^i_{s+1,\eta-1}(k-\mu)\right)^T \mathbf{\Lambda} \mathbf{z}^i_{kM-s}}{\beta^i_{s+1,\eta-1}(k)} \mathbf{b}^i_{s+1,\eta-1}(k)
\end{aligned}
$$

$$
\begin{aligned}
&= \mathbf{f}^i_{s,\eta-1}(k) + \frac{\left(\mathbf{b}^i_{s+1,\eta-1}(k-\mu)\right)^T \mathbf{\Lambda}\left(\mathbf{z}^i_{kM-s} - \mathbf{f}^i_{s,\eta-1}(k)\right)}{\beta^i_{s+1,\eta-1}(k)}\mathbf{b}^i_{s+1,\eta-1}(k) \\
&\quad - \frac{\left(\mathbf{b}^i_{s+1,\eta-1}(k-\mu)\right)^T \mathbf{\Lambda}\mathbf{z}^i_{kM-s}}{\beta^i_{s+1,\eta-1}(k)}\mathbf{b}^i_{s+1,\eta-1}(k) \\
&= \mathbf{f}^i_{s,\eta-1}(k) - \frac{\left(\mathbf{b}^i_{s+1,\eta-1}(k-\mu)\right)^T \mathbf{\Lambda}\mathbf{f}^i_{s,\eta-1}(k)}{\beta^i_{s+1,\eta-1}(k)}\mathbf{b}^i_{s+1,\eta-1}(k) \\
&= \mathbf{f}^i_{s,\eta-1}(k) - \frac{\psi^i_{s,\eta}(k)}{\beta^i_{s+1,\eta-1}(k)}\mathbf{b}^i_{s+1,\eta-1}(k)
\end{aligned}
$$

where on the right hand side (RHS), in addition to the first and second equations, (1) the third equation is derived by using the orthogonality between $\mathbf{\Lambda}\mathbf{b}^i_{s+1,\eta-1}(k)$ and $\mathbf{h}^i_{s,\eta}(k-\mu)$ and the definition of $\beta^i_{s+1,\eta-1}(k)$; (2) the fourth equation is reached in terms of the formula

$$
\begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{B} & \mathbf{C} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{A}^{-1} & \mathbf{0} \\ -\mathbf{C}^{-1}\mathbf{B}\mathbf{A}^{-1} & \mathbf{C}^{-1} \end{bmatrix}
$$

for any non-singular matrices $\mathbf{A}$ and $\mathbf{C}$ and a straightforward manipulation; (3) the definition of $\mathbf{f}^i_{s,\eta}(k)$ is employed twice in the derivation of the fifth equation; and (4) the final equation arrives by the use of the definition of $\psi^i_{s,\eta}(k)$.

A.2 Derivation of Equation 3.39

The derivation of Equation 3.39 is similar to that of Equation 3.37. From the definition of $\mathbf{b}^i_{s,\eta}(k)$ and Equation 3.24 it follows that

$$
\begin{aligned}
\mathbf{b}^i_{s,\eta}(k) &= \mathbf{z}^i_{kM-s-\eta} - \mathbf{z}^i_{kM-s-\eta}|_{\mathbf{h}^i_{s,\eta-1}(k)\oplus \mathrm{Span}\{\mathbf{f}^i_{s,\eta-1}(k)\}} \\
&= \mathbf{z}^i_{kM-s-\eta} - \left[\mathbf{z}^i_{kM-s-1:kM-s-\eta+1} \;\; \mathbf{f}^i_{s,\eta-1}(k)\right] \bullet \\
&\quad \left(\begin{bmatrix} \left(\mathbf{z}^i_{(k-\mu)M-s-1:(k-\mu)M-s-\eta+1}\right)^T \\ \left(\mathbf{f}^i_{s,\eta-1}(k-\mu)\right)^T \end{bmatrix} \mathbf{\Lambda} \left[\mathbf{z}^i_{kM-s-1:kM-s-\eta+1} \;\; \mathbf{f}^i_{s,\eta-1}(k)\right] \right)^{-1} \bullet \\
&\quad \begin{bmatrix} \left(\mathbf{z}^i_{(k-\mu)M-s-1:(k-\mu)M-s-\eta+1}\right)^T \mathbf{\Lambda} \mathbf{z}^i_{kM-s-\eta} \\ \left(\mathbf{f}^i_{s,\eta-1}(k-\mu)\right)^T \mathbf{\Lambda} \mathbf{z}^i_{kM-s-\eta} \end{bmatrix} \\
&= \mathbf{z}^i_{kM-s-\eta} - \left[\mathbf{z}^i_{kM-s-1:kM-s-\eta+1} \;\; \mathbf{f}^i_{s,\eta-1}(k)\right] \bullet \\
&\quad \begin{bmatrix} \left(\mathbf{z}^i_{(k-\mu)M-s-1:(k-\mu)M-s-\eta+1}\right)^T \mathbf{\Lambda} \mathbf{z}^i_{kM-s-1:kM-s-\eta+1} & 0 \\ \left(\mathbf{f}^i_{s,\eta-1}(k-\mu)\right)^T \mathbf{\Lambda} \mathbf{z}^i_{kM-s-1:kM-s-\eta+1} & \alpha^i_{s,\eta-1}(k) \end{bmatrix}^{-1} \bullet \\
&\quad \begin{bmatrix} \left(\mathbf{z}^i_{(k-\mu)M-s-1:(k-\mu)M-s-\eta+1}\right)^T \mathbf{\Lambda} \mathbf{z}^i_{kM-s-\eta} \\ \left(\mathbf{f}^i_{s,\eta-1}(k-\mu)\right)^T \mathbf{\Lambda} \mathbf{z}^i_{kM-s-\eta} \end{bmatrix} \\
&= \mathbf{z}^i_{kM-s-\eta} - \mathbf{z}^i_{kM-s-1:kM-s-\eta+1} \left(\left(\mathbf{z}^i_{(k-\mu)M-s-1:(k-\mu)M-s-\eta+1}\right)^T \mathbf{\Lambda} \mathbf{z}^i_{kM-s-1:kM-s-\eta+1} \right)^{-} \\
&\quad \left(\mathbf{z}^i_{(k-\mu)M-s-1:(k-\mu)M-s-\eta+1}\right)^T \mathbf{\Lambda} \mathbf{z}^i_{kM-s-\eta} \\
&\quad + \frac{\mathbf{f}^i_{s,\eta-1}(k)}{\alpha^i_{s,\eta-1}(k)} \left(\mathbf{f}^i_s(k-\mu)\right)^T \mathbf{\Lambda} \mathbf{z}^i_{kM-s-1:kM-s-\eta+1} \bullet \\
&\quad \left(\left(\mathbf{z}^i_{(k-\mu)M-s-1:(k-\mu)M-s-\eta+1}\right)^T \mathbf{\Lambda} \mathbf{z}^i_{kM-s-1:kM-s-\eta+1} \right)^{-1} \bullet \\
&\quad \left(\mathbf{z}^i_{(k-\mu)M-s-1:(k-\mu)M-s-\eta+1}\right)^T \mathbf{\Lambda} \mathbf{z}^i_{kM-s-\eta} - \frac{\left(\mathbf{f}^i_{s,\eta-1}(k-\mu)\right)^T \mathbf{\Lambda} \mathbf{z}^i_{kM-s-\eta}}{\alpha^i_{s,\eta-1}(k)} \mathbf{f}^i_{s,\eta-1}(k) \\
&= \mathbf{b}^i_{s+1,\eta-1}(k) + \frac{\left(\mathbf{f}^i_{s,\eta-1}(k-\mu)\right)^T \mathbf{\Lambda} \left(\mathbf{z}^i_{kM-s-\eta} - \mathbf{b}^i_{s+1,\eta-1}(k)\right)}{\alpha^i_{s,\eta-1}(k)} \mathbf{f}^i_{s,\eta-1}(k) \\
&\quad - \frac{\left(\mathbf{f}^i_{s,\eta-1}(k-\mu)\right)^T \mathbf{\Lambda} \mathbf{z}^i_{kM-s-\eta}}{\alpha^i_{s,\eta-1}(k)} \mathbf{f}^i_{s,\eta-1}(k) \\
&= \mathbf{b}^i_{s+1,\eta-1}(k) - \frac{\varpi^i_{s,\eta}(k)}{\alpha^i_{s,\eta-1}(k)} \mathbf{f}^i_{s,\eta-1}(k)
\end{aligned}
$$

where the definitions of $\alpha^i_{s,\eta}(k)$ and $\varpi^i_{s,\eta}(k)$ have been used.

A.3 Derivation of Equations 3.43 and 3.44

In order to prove Equations 3.43 and 3.44, in addition to Equation 3.42, there is a need to derive another order-recursion formula for $\boldsymbol{\phi}_{\mathbf{h}^i_{s-1,\eta+1}(k)}$. Based on Equations 3.16 and 3.18,

the subspace $\mathbf{h}^i_{s-1,\eta+1}(k)$ can be decomposed into

$$\mathbf{h}^i_{s-1,\eta+1}(k) = \mathbf{h}^i_{s,\eta}(k) \oplus \mathbf{f}^i_{s,\eta}(k)$$

Subsequently, from the non-symmetric projection formula given by Equation 3.24,

$$\begin{aligned}
\boldsymbol{\phi}_{\mathbf{h}^i_{s-1,\eta+1}(k)} &= \boldsymbol{\phi} - \boldsymbol{\phi}|_{\mathbf{h}^i_{s,\eta}(k) \oplus \mathbf{f}^i_{s,\eta}(k)} \\
&= \boldsymbol{\phi} - \left[\mathbf{z}^i_{kM-s-1:kM-s-\eta} \;\; \mathbf{f}^i_{s,\eta}(k)\right] \bullet \\
&\quad \begin{bmatrix} \left(\mathbf{z}^i_{(k-\mu)M-s-1:(k-\mu)M-s-\eta}\right)^T \boldsymbol{\Lambda} \mathbf{z}^i_{kM-s-1:kM-s-\eta} & 0 \\ \left(\mathbf{f}^i_{s,\eta}(k-\mu)\right)^T \boldsymbol{\Lambda} \mathbf{z}^i_{kM-s-1:kM-s-\eta} & \alpha^i_{s,\eta}(k) \end{bmatrix}^{-1} \bullet \\
&\quad \begin{bmatrix} \left(\mathbf{z}^i_{(k-\mu)M-s-1:(k-\mu)M-s-\eta}\right)^T \boldsymbol{\Lambda} \boldsymbol{\phi} \\ \left(\mathbf{f}^i_{s,\eta}(k-\mu)\right)^T \boldsymbol{\Lambda} \boldsymbol{\phi} \end{bmatrix} \\
&= \boldsymbol{\phi} - \mathbf{z}^i_{kM-s-1:kM-s-\eta} \bullet \\
&\quad \left(\left(\mathbf{z}^i_{(k-\mu)M-s-1:(k-\mu)M-s-\eta}\right)^T \boldsymbol{\Lambda} \mathbf{z}^i_{kM-s-1:kM-s-\eta}\right)^{-1} \left(\mathbf{z}^i_{(k-\mu)M-s-1:(k-\mu)M-s-\eta}\right)^T \\
&\quad + \mathbf{f}^i_{s,\eta}(k) \frac{\left(\mathbf{f}^i_{s,\eta}(k-\mu)\right)^T \boldsymbol{\Lambda} \left(\boldsymbol{\phi} - \boldsymbol{\phi}_{\mathbf{h}^i_{s,\eta}(k)}\right)}{\alpha^i_{s,\eta}(k)} - \mathbf{f}^i_{s,\eta}(k) \frac{\left(\mathbf{f}^i_{s,\eta}(k-\mu)\right)^T \boldsymbol{\Lambda} \boldsymbol{\phi}}{\alpha^i_{s,\eta}(k)} \\
&= \boldsymbol{\phi}_{\mathbf{h}^i_{s,\eta}(k)} - \mathbf{f}^i_{s,\eta}(k) \frac{\left(\mathbf{f}^i_{s,\eta}(k-\mu)\right)^T \boldsymbol{\Lambda} \boldsymbol{\phi}_{\mathbf{h}^i_{s,\eta}(k)}}{\alpha^i_{s,\eta}(k)} \\
&= \boldsymbol{\phi}_{\mathbf{h}^i_{s,\eta}(k)} - \mathbf{f}^i_{s,\eta}(k) \frac{\epsilon^i_{s,\eta}(k)}{\alpha^i_{s,\eta}(k)}
\end{aligned}$$

where $\boldsymbol{\Lambda}\mathbf{f}^i_{s,\eta}(k) \perp \mathbf{h}^i_{s,\eta}(k-\mu)$, the identity for computing the inverse of a composite matrix given at the end of Appendix A, and the definitions of $\boldsymbol{\phi}_{\mathbf{h}^i_{s,\eta}(k)}$, $\alpha^i_{s,\eta}(k)$ and $\epsilon^i_{s,\eta}(k)$ have been utilized.

A.3.1 Derivation of Equation 3.43

The substitution of Equation 3.39 and the following equation

$$\boldsymbol{\phi}_{\mathbf{h}^i_{s-1,\eta+1}(k)} = \boldsymbol{\phi}_{\mathbf{h}^i_{s,\eta}(k)} - \mathbf{f}^i_{s,\eta}(k) \frac{\epsilon^i_{s,\eta}(k)}{\alpha^i_{s,\eta}(k)}$$

into Equation 3.29 gives

$$\begin{aligned}
\varsigma^i_{s,\eta}(k) &= < \mathbf{b}^i_{s,\eta}(k-\mu), \boldsymbol{\phi}_{\mathbf{h}^i_{s-1,\eta}(k)} > \\
&= < \mathbf{b}^i_{s+1,\eta-1}(k-\mu) - \frac{\varpi^i_{s,\eta}(k-\mu)}{\alpha^i_{s,\eta-1}(k-\mu)} \mathbf{f}^i_{s,\eta-1}(k-\mu), \\
&\quad \boldsymbol{\phi}_{\mathbf{h}^i_{s,\eta-1}(k)} - \mathbf{f}^i_{s,\eta-1}(k) \frac{\epsilon^i_{s,\eta-1}(k)}{\alpha^i_{s,\eta-1}(k)} > \\
&= < \mathbf{b}^i_{s+1,\eta-1}(k-\mu), \boldsymbol{\phi}_{\mathbf{h}^i_{s,\eta-1}(k)} > - \frac{\epsilon^i_{s,\eta-1}(k)}{\alpha^i_{s,\eta-1}(k)} < \mathbf{b}^i_{s+1,\eta-1}(k-\mu), \mathbf{f}^i_{s,\eta-1}(k) > \\
&= \varsigma^i_{s+1,\eta-1}(k) - \frac{\psi^i_{s,\eta}(k) \epsilon^i_{s,\eta-1}(k)}{\alpha^i_{s,\eta-1}(k)}
\end{aligned}$$

where the definitions of $\alpha^i_{s,\eta}(k)$, $\psi^i_{s,\eta}(k)$ and $\epsilon^i_{s,\eta}(k)$ have been considered. Similarly, it can be proven that

$$\varsigma^i_{M,\eta}(k) = \varsigma^i_{1,\eta-1}(k-1) - \frac{\epsilon^i_{M,\eta-1}(k)\psi^i_{M,\eta}(k)}{\alpha^i_{M,\eta-1}(k)}$$

A.3.2 Derivation of Equation 3.44

The substitution of Equations 3.37 and 3.42 into Equation 3.28 leads to

$$\begin{aligned}
\epsilon^i_{s,\eta}(k) &= < \mathbf{f}^i_{s,\eta}(k-\mu), \boldsymbol{\phi}_{\mathbf{h}^i_{s,\eta}(k)} > \\
&= < \mathbf{f}^i_{s,\eta-1}(k-\mu) - \frac{\psi^i_{s,\eta}(k-\mu)}{\beta^i_{s+1,\eta-1}(k-\mu)}\mathbf{b}^i_{s+1,\eta-1}(k-\mu), \\
&\quad \boldsymbol{\phi}_{\mathbf{h}^i_{s,\eta-1}(k)} - \mathbf{b}^i_{s+1,\eta-1}(k)\frac{\varsigma^i_{s+1,\eta-1}(k)}{\beta^i_{s+1,\eta-1}(k)} > \\
&= < \mathbf{f}^i_{s,\eta-1}(k-\mu), \boldsymbol{\phi}_{\mathbf{h}^i_{s,\eta-1}(k)} > - \frac{\varsigma^i_{s+1,\eta-1}(k)}{\beta^i_{s+1,\eta-1}(k)} < \mathbf{f}^i_{s,\eta-1}(k-\mu), \mathbf{b}^i_{s+1,\eta-1}(k) > \\
&= \epsilon^i_{s,\eta-1}(k) - \frac{\varpi^i_{s,\eta}(k)\varsigma^i_{s+1,\eta-1}(k)}{\beta^i_{s+1,\eta-1}(k)}, \; s = 1 \cdots M-1.
\end{aligned}$$

where we have used the definitions of $\epsilon^i_{s,\eta}(k)$, $\varpi^i_{s,\eta}(k)$, and $\psi^i_{s,\eta}(k)$. Similarly, we have

$$\epsilon^i_{s,\eta}(k) = \epsilon^i_{s,\eta-1}(k) - \frac{\varpi^i_{s,\eta}(k)\varsigma^i_{1,\eta-1}(k-1)}{\beta^i_{1,\eta-1}(k-1)}, \; s = M.$$

A.4 Proof of Equation 3.46

According to Equation 3.25,

$$\begin{aligned}
\mathbf{h}_\eta(k) \oplus \boldsymbol{\phi} &= (\mathbf{h}_\eta(k))_{\boldsymbol{\phi}} \oplus \mathbf{h}_\eta(k)|_{\boldsymbol{\phi}} \oplus \boldsymbol{\phi} \\
&= (\mathbf{h}_\eta(k))_{\boldsymbol{\phi}} \oplus \boldsymbol{\phi}
\end{aligned}$$

Note that

$$(\mathbf{h}_\eta(k))_{\boldsymbol{\phi}} \perp \boldsymbol{\phi}$$

because $\boldsymbol{\phi}$ is a constant vector. Therefore,

$$\mathbf{h}_\eta(k) \oplus \boldsymbol{\phi} = (\mathbf{h}_\eta(k))_{\boldsymbol{\phi}} \overset{\perp}{\oplus} \boldsymbol{\phi}$$

where $\overset{\perp}{\oplus}$ denotes the direct sum of two subspaces. Subsequently,

$$\begin{aligned}
(\mathbf{a}_k)_{(\mathbf{h}_\eta(k) \oplus \boldsymbol{\phi})} &= \mathbf{a}_k - (\mathbf{a}_k)|_{\boldsymbol{\phi}} - (\mathbf{a}_k)|_{\left((\mathbf{h}_\eta(k))_{\boldsymbol{\phi}}\right)} \\
&= (\mathbf{a}_k)_{\boldsymbol{\phi}} - \left((\mathbf{a}_k)|_{\boldsymbol{\phi}} + (\mathbf{a}_k)_{\boldsymbol{\phi}}\right)|_{\left((\mathbf{h}_\eta(k))_{\boldsymbol{\phi}}\right)} \\
&= \left((\mathbf{a}_k)_{\boldsymbol{\phi}}\right)_{\left((\mathbf{h}_\eta(k))_{\boldsymbol{\phi}}\right)}
\end{aligned}$$

where Equation 3.25 and the orthogonality between $(\mathbf{a}_k)|_{\boldsymbol{\phi}}$ and $(\mathbf{h}_\eta(k))_{\boldsymbol{\phi}}$ have been used.

A.5 The Complete Algorithm for the Recursive Identification of $\Theta_n^i(k)$

The complete lattice filter algorithm for the recursive identification of $\boldsymbol{\theta}_n^1$ is listed as follows, where unless otherwise stated, $s = 1 \cdots M$, $\eta = 1 \cdots M_n - 1$, and $k >= 2\mu + Int\left(\frac{s+\eta-1}{M}\right)$.

$$\varepsilon_{s,0}^i(k) = \gamma_{s,0}^i(k) = z^i(kM - s),\ k > 0.$$

$$\epsilon_{s,0}^i(k) = \varsigma_{s,0}^i(k) = z^i((k-\mu)M - s),\ k > \mu.$$

$$\alpha_{s,0}^i(k) = \beta_{s,0}^i(k) = \sum_{t=-\infty}^{k} \lambda^{k-t} z^i(tM - s) z^i((t-\mu)M - s);\ z^i(\tau) = 0,\ \text{if } \tau < 0.$$

$$\varpi_{s,\eta}^i(k) = \begin{cases} \lambda \varpi_{s,\eta}^i(k-1) + \frac{\epsilon_{s,\eta-1}^i(k)\gamma_{s+1,\eta-1}^i(k)}{1-\pi_{s+1,\eta-1}^i(k)}, & s \neq M. \\ \lambda \varpi_{s,\eta}^i(k-1) + \frac{\epsilon_{s,\eta-1}^i(k)\gamma_{1,\eta-1}^i(k-1)}{1-\pi_{1,\eta-1}^i(k-1)}, & s = M. \end{cases}$$

$$\psi_{s,\eta}^i(k) = \begin{cases} \lambda \psi_{s,\eta}^i(k-1) + \frac{\varsigma_{s+1,\eta-1}^i(k)\epsilon_{s,\eta-1}^i(k)}{1-\pi_{s+1,\eta-1}^i(k)} & s \neq M. \\ \lambda \psi_{s,\eta}^i(k-1) + \frac{\varsigma_{1,\eta-1}^i(k-1)\epsilon_{s,\eta-1}^i(k)}{1-\pi_{1,\eta-1}^i(k-1)} & s = M. \end{cases}$$

$$\varepsilon_{s,\eta}^i(k) = \begin{cases} \varepsilon_{s,\eta-1}^i(k) - \frac{\psi_{s,\eta}^i(k)\gamma_{s+1,\eta-1}^i(k)}{\beta_{s+1,\eta-1}^i(k)}, & s \neq M. \\ \varepsilon_{s,\eta-1}^i(k) - \frac{\psi_{s,\eta}^i(k)\gamma_{1,\eta-1}^i(k-1)}{\beta_{1,\eta-1}^i(k-1)}, & s = M. \end{cases}$$

$$\gamma_{s,\eta}^i(k) = \begin{cases} \gamma_{s+1,\eta-1}^i(k) - \frac{\varpi_{s,\eta}^i(k)\epsilon_{s,\eta-1}^i(k)}{\alpha_{s,\eta-1}^i(k)}, & s \neq M. \\ \gamma_{1,\eta-1}^i(k-1) - \frac{\varpi_{s,\eta}^i(k)\epsilon_{s,\eta-1}^i(k)}{\alpha_{s,\eta-1}^i(k)}, & s = M. \end{cases}$$

$$\alpha_{s,\eta}^i(k) = \begin{cases} \alpha_{s,\eta-1}^i(k) - \frac{\varpi_{s,\eta}^i(k)\psi_{s,\eta}^i(k)}{\beta_{s+1,\eta-1}^i(k)}, & s \neq M. \\ \alpha_{s,\eta-1}^i(k) - \frac{\varpi_{s,\eta}^i(k)\psi_{s,\eta}^i(k)}{\beta_{1,\eta-1}^i(k-1)}, & s = M. \end{cases}$$

$$\beta_{s,\eta}^i(k) = \begin{cases} \beta_{s+1,\eta-1}^i(k) - \frac{\psi_{s,\eta}^i(k)\varpi_{s,\eta}^i(k)}{\alpha_{s,\eta-1}^i(k)}, & s \neq M. \\ \beta_{1,\eta-1}^i(k-1) - \frac{\psi_{s,\eta}^i(k)\varpi_{s,\eta}^i(k)}{\alpha_{s,\eta-1}^i(k)}, & s = M. \end{cases}$$

$$\epsilon_{s,\eta}^i(k) = \begin{cases} \epsilon_{s,\eta-1}^i(k) - \frac{\varpi_{s,\eta}^i(k)\varsigma_{s+1,\eta-1}^i(k)}{\beta_{s+1,\eta-1}^i(k)} & s \neq M. \\ \epsilon_{s,\eta-1}^i(k) - \frac{\varpi_{s,\eta}^i(k)\varsigma_{1,\eta-1}^i(k-1)}{\beta_{1,\eta-1}^i(k-1)} & s = M. \end{cases}$$

$$\varsigma_{s,\eta}^i(k) = \begin{cases} \varsigma_{s+1,\eta-1}^i(k) - \frac{\psi_{s,\eta}^i(k)\epsilon_{s,\eta-1}^i(k)}{\alpha_{s,\eta-1}^i(k)} & s \neq M. \\ \varsigma_{1,\eta-1}^i(k-1) - \frac{\psi_{s,\eta}^i(k)\epsilon_{s,\eta-1}^i(k)}{\alpha_{s,\eta-1}^i(k)} & s = M. \end{cases}$$

$$\pi^i_{s,\eta}(k) = \pi^i_{s,\eta-1}(k) + \frac{\gamma^i_{s,\eta-1}(k)\zeta^i_{s,\eta-1}(k)}{\beta^i_{s,\eta-1}(k)}, \text{ with } \pi^i_{s,0}(k) = 0.$$

$$\pi^i_{s,\eta}(k) = \begin{cases} \pi^i_{s+1,\eta-1}(k) + \frac{\epsilon^i_{s,\eta-1}(k)\epsilon^i_{s,\eta-1}(k)}{\alpha^i_{s,\eta-1}(k)} & s \neq M. \\ \pi^i_{1,\eta-1}(k-1) + \frac{\epsilon^i_{s,\eta-1}(k)\epsilon^i_{s,\eta-1}(k)}{\alpha^i_{s,\eta-1}(k)} & s = M. \end{cases}$$

$$\alpha^i_{s,\eta}(k) = \begin{cases} \lambda\alpha^i_{s,\eta}(k-1) + \frac{\epsilon^i_{s,\eta}(k)\epsilon^i_{s,\eta}(k)}{1-\pi^i_{s+1,\eta}(k)}, & s \neq M.\ \eta = 0 \cdots M_n - 1. \\ \lambda\alpha^i_{s,\eta}(k-1) + \frac{\epsilon^i_{s,\eta}(k)\epsilon^i_{s,\eta}(k)}{1-\pi^i_{1,\eta}(k-1)}, & s = M.\ \eta = 0 \cdots M_n - 1. \end{cases}$$

$$\beta^i_{s,\eta}(k) = \lambda\beta^i_{s,\eta}(k-1) + \frac{\gamma^i_{s,\eta}(k)\zeta^i_{s,\eta}(k)}{1-\pi^i_{s,\eta}(k)}, \ \eta = 0 \cdots M_n - 1.$$

$$\vartheta^i_{s,\eta}(k) = \begin{cases} \left[\left(\vartheta^i_{s,\eta-1}(k)\right)^T \mid 0\right]^T - \frac{\psi^i_{s,\eta}(k)}{\beta^i_{s+1,\eta-1}(k)}\left[\left(\varrho^i_{s+1,\eta-1}(k)\right)^T \mid -1\right]^T, & s \neq M. \\ \left[\left(\vartheta^i_{s,\eta-1}(k)\right)^T \mid 0\right]^T - \frac{\psi^i_{s,\eta}(k)}{\beta^i_{1,\eta-1}(k-1)}\left[\left(\varrho^i_{1,\eta-1}(k-1)\right)^T \mid -1\right]^T, & s = M. \end{cases}$$

$$\varrho^i_{s,\eta}(k) = \begin{cases} \left[0 \mid \left(\varrho^i_{s+1,\eta-1}(k)\right)^T\right]^T - \frac{\varpi^i_{s,\eta}(k)}{\alpha^i_{s,\eta-1}(k)}\left[-1 \mid \left(\vartheta^i_{s,\eta-1}(k)\right)^T\right]^T, & s \neq M. \\ \left[0 \mid \left(\varrho^i_{1,\eta-1}(k-1)\right)^T\right]^T - \frac{\varpi^i_{s,\eta}(k)}{\alpha^i_{s,\eta-1}(k)}\left[-1 \mid \left(\vartheta^i_{s,\eta-1}(k)\right)^T\right]^T, & s = M. \end{cases}$$

$$\varrho^i_{s,0}(k) = \vartheta^i_{s,0}(k) = 0.$$

Note that to compute the parameters

$$\Theta_n(k) \equiv \begin{bmatrix} \theta^1_n(k) \\ \vdots \\ \theta^m_n(k) \end{bmatrix}$$

the above algorithm should be executed from $i = 1$ until $i = m$, and taking into account the initialization procedure given before.